Universidad de Deusto
Deustuko Unibertsitatea
Deusto

Máster Universitario en
Seguridad de la Información
2012-2013

Proyecto Fin de Máster

“Modelos de Inteligencia Computacional aplicados al diseño e investigación de Sistema de Detección de Intrusos.”

Autor:

Ing. Arthur Christian Huamani Cuba.

Director:
Dr. Emilio Corchado Rodríguez

Bilbao, junio de 2013

A mis padres. Por su esfuerzo, dedicación y cariño.

Resumen

El imparable crecimiento de internet en el mundo, y la integración de los diversos sistemas informáticos conectados a la red de Internet han hecho que la Seguridad de la Información alcance un papel de suma importancia actualmente. Desde el momento en que un equipo (bien un ordenador, una tableta o un "*smartphone*") se conecta a Internet, se abren toda una serie de posibilidades; sin embargo, éstas traen consigo nuevos y en ocasiones complejos tipos de ataques (CHIRAG, 2013). Para que las informaciones sean protegidas de acciones maliciosas, es necesario hacer uso de herramientas especializadas en la protección de datos e informaciones, como antivirus, antispam, firewall, IDS, IPS, balanceadores de cargas, control de accesos, sistemas criptográficos tipos VPN y entre otras. Un sistema que auxilia y automatiza la protección de las informaciones es el sistema de detección de intrusos (IDS) que tiene como intensión de detectar el uso no autorizado, ataques en ordenadores o en redes de ordenadores. Sin embargo, algunas funcionalidades de esos sistemas son limitadas, tales como la detección, el análisis y la respuesta instantánea de un nuevo ataque (*Zero day attack*) (CHRISTOPHER, 2013).

La aplicación de técnicas basadas en Inteligencia Artificial (AI) para la detección de intrusos (IDS) fundamentalmente las redes neuronales artificiales (ANN), están demostrando ser un enfoque muy adecuado para paliar muchos de los problemas que se dan en esta área. Sin embargo, gran volumen de información que se requiere cada día para entrenar estos sistemas, junto con la necesidad exponencial de tiempo que requieren para asimilarlos, dificulta enormemente su puesta en marcha en escenarios reales. (SHELLY, 2010).

En el presente proyecto de fin de máster se propone el diseño de un sistema capaz de detectar situaciones anómalas en la red, basado en modelos de inteligencia computacional e instrumento complementario para visualizar la estructura interior de un conjunto de datos asociados al tráfico de redes, se aplicarán diversas técnicas EPP (*Exploratory Proyection Pursuit*) como PCA (*Principal Component Analysis*), MLHL (*Maximum Likelihood Hebbian Learning*) y CMLH (*Cooperative Maximum Likelihood Hebian Learning*), para analizar la estructura interna del conjunto de datos y de este modo identificar

comportamiento anómalos asociados a situaciones de riesgo en la red (CORCHADO, HERRERO, 2013).

.

El presente proyecto de fin de máster se centrará principalmente en el enfoque de detección de anomalías centrándose en la capacidad de generalización de algunas herramientas basadas en aprendizaje no supervisado, estudiando la posibilidad de detectar también nuevos ataques (*Zero day attack*).

Para validar la propuesta se ha diseñado un escenario de prueba mediante un “IDS basado en aprendizaje no supervisado”. Los resultados obtenidos a partir de las pruebas realizadas demuestran la validez de la propuesta y acreditan las líneas futuras de trabajo.

Áreas del Proyecto: Sistema de Detección de Intrusos, Red Neuronal Artificial, Captura del Tráfico de Red, Neural IDS, Aprendizaje No Supervisado, Seguridad de la Información, Detección de Anomalías, Modelos de Redes Probabilísticas.

Summary

The relentless growth of Internet in the world, and the integration of different computer systems connected to the Internet have made information security reach a critical role today. From the moment a computer (such as a laptop, a tablet or a smartphone") is connected to the Internet, open a range of possibilities, but they bring new and sometimes complex types of attacks (CHIRAG, 2013). For that information is protected from malicious actions, it is necessary to use specialized tools to protect data and information such as antivirus, anti-spam, firewall, IDS, IPS, load balancers, access control, cryptographic systems VPN types and among others. A system that automates and helps protect information is the intrusion detection system (IDS) whose intention to detect unauthorized use, strikes on computers or computer networks. However, some features of these systems are limited, such as the detection, analysis and instant response of a new attack (Zero day attack) (CHRISTOPHER, 2013).

The application of techniques based on Artificial Intelligence (AI) for intrusion detection (IDS), mainly artificial neural networks (ANN), are proving to be a very suitable approach to overcome many of the problems that occur in this area. However, a large volume of information that is required every day to train these systems, along with the need exponential time required to assimilate, greatly hampers its implementation in real scenarios (SHELLY, 2010).

In this master's thesis project proposes the design of a system capable of detecting malfunctions in the network, based on computational intelligence models and complementary tool to visualize the internal structure of a data set associated with network traffic is apply various techniques EPP (Exploratory Projection Pursuit) and PCA (Principal Component Analysis), MLHL (Maximum Likelihood Learning Hebbian) and CMLH (Maximum Likelihood Hebbian Cooperative Learning), to analyze the internal structure of the data set and thus identify behavior associated anomalous situations on the network risk (CORCHADO, HERRERO, 2013).

This Master Thesis project will mainly focus on the anomaly detection approach focusing on the generalization of some tools based on unsupervised learning, considering also detect new attacks (Zero day attack).

To validate the proposal we have designed a test scenario using a “unsupervised learning-based IDS”. The results obtained from tests show the validity of the proposal and accredit future lines of work.

Project Areas: Intrusion Detection System, Artificial Neural Network, Network Traffic Capture, Neural IDS, Unsupervised Learning, Information Security, Anomaly Detection, Probabilistic Network Models.

Agradecimientos

En estas primeras líneas de redacción a mi proyecto de fin de máster, quiero empezar dando gracias a mis padres: [illegible] que son mi apoyo y mi fuerza para continuar luchando y así ir alcanzando mis objetivos trazados; y a mis hermanos: [illegible] por su preocupación y cariño.

[illegible]

A mi tío Jesús por su inmenso cariño a la familia, por su incondicional apoyo y ser parte de mis proyectos futuros que a pesar de la distancia, él siempre está con nosotros.

A mi director del proyecto de fin de máster, Dr. Emilio Corchado Rodríguez; por abrirme las puertas de su laboratorio, en la Universidad de Salamanca, en provecho a la realización de la presente investigación, por su orientación, por sus consejos como amigo y por su confianza depositada en mí. Gracias igualmente al Ing. Héctor Quintián Pardo, por su apoyo constante e incondicional dedicando inclusive parte de su tiempo libre para poder responder a mis interrogantes en cada una de las etapas para la realización de mi proyecto.

A mis profesores del MUSI, por haberme brindando sus conocimientos y experiencias. Pero en especial al Dr. Carlos Cazorla por sus sabios consejos, por su gran amistad y preocupación para con mi persona.

A mis compañeros del MUSI: Javier, Carlos, Mohammed, Lander, Juan, Manuel, Igor, Adrian, Josu, Samuel, Borja y Galder con quienes compartí alegrías, enojos, preocupaciones y muchas cosas más que quedarán por siempre en mi mente y en mi

corazón.

Gracias verdaderamente a cada uno de ustedes, porque han hecho de mi estancia en España “la mejor” y haber sido parte activa de uno de mis grandes objetivos. ¡PASCHI!

Tabla de Contenido

Tabla de Figuras

Índice de Tablas

Abreviaturas

AI	Artificial Intelligence.
ANN	Artificial Neural Network.
BDI	Belief, Desire and Intention.
CBR	Case-Based Reasoning.
CCA	Curvilinear Component Analysis.
CMLHL	Cooperative Maximum-Likelihood Hebbian Learning.
EPP	Exploratory Projection Pursuit.
HIDS	Host-Based Intrusion Detection System.
ID	Intrusion Detection.
IDS	Intrusion Detection System.
IPS	Intrusion Prevention System.
IP	Internet Protocol.
KNN	Khonen Neural Network.
LAN	Local Area Network.
MAS	Multiagent System.
MIB	Management Information Base.
MIT	Massachusetts Institute of Technology.
MLHL	Maximum-Likelihood Hebbian Learning.

NFN	Negative Feedback Network.
NID	Network-Based Intrusion Detection.
NIDS	Network-Based Intrusion Detection System.
PCA	Principal Component Analysis.
PFM	Project of Master's degree.
RFC	Request For Comments.
SNMP	Simple Network Management Protocol.
SNNS	Stuttgart Neural Network Simulator.
SOM	Self-Organizing Map.
SVM	Support Vector Machine.
TCP	Transmission Control Protocol.
WSI	Institute Wilhelm Schickard.

Si conoces a los demás y te conoces a ti mismo, ni en cien batallas correrás peligro; si no conoces a los demás, pero te conoces a ti mismo, perderás una batalla y ganarás otra; si no conoces a los demás ni te conoces a ti mismo, correrás peligro en cada batalla.

SUN TZU, 孫子 (Reino de Wu, Siglo VI a.C.), El Arte de la Guerra.

INTRODUCCIÓN

"SER LO QUE SOY, NO ES NADA SIN LA SEGURIDAD".

W. SHAKESPEARE (1564-1616)

Sin duda W. Shakespeare tenía un concepto más evolucionado de la seguridad que sus contemporáneos del siglo XV y quizás también que algunos de los nuestros.

La meta es ambiciosa, la seguridad como materia académica no existe, y es considerada por los "estudiosos" como una herramienta dentro del ámbito en que se la estudia: relaciones internacionales-nacionales, estudios de riesgo, prevención de crímenes y pérdidas, etc. Muchos sostienen que es una teoría tan amplia, compleja y abstracta como la pobreza, la belleza o el amor; y ni siquiera arriesgan su definición. El amplio desarrollo de las nuevas tecnologías informáticas está ofreciendo un nuevo campo de acción a conductas antisociales y delictivas manifestadas en formas antes imposibles de imaginar, ofreciendo la posibilidad de cometer delitos tradicionales en formas no tradicionales (BORGHELLO, 2001).

Durante los últimos años las intrusiones y los ataques informáticos a través de Internet se han incrementado notablemente. Este incremento en el número de incidentes ha venido acompañado por una clara evolución de las herramientas y técnicas utilizadas por los atacantes. Parte de los responsables de estas acciones son usuarios avanzados que desarrollan sus propias aplicaciones y son capaces de crear y utilizar sofisticadas puertas traseras para introducirse en otros sistemas. Estos individuos son los más cercanos a la figura del Hacker [1]. (SPITZNER, 2008)

Esta idea generalizada y en muchos casos mitificada sobre el perfil de estos intrusos hace pensar que sólo serán objeto de ataque aquellos equipos que contengan información trascendente. Sin embargo, esto es un grave error. Estos ataques selectivos dirigidos por expertos suponen un porcentaje muy pequeño de los que a diario se producen a través de la red de Internet. (CHRISTOPHER, 2013) En la práctica, la totalidad de los incidentes que acontecen en Internet no van dirigidos contra equipos ni compañías específicas, sino que tienen como objetivo a la víctima fácil. El blanco seleccionado puede ser cualquier equipo conectado a la red de Internet que posea una debilidad específica que el atacante busca y es capaz de aprovechar para conseguir el acceso a la máquina.

Según la Unión Internacional de Telecomunicaciones (ITU)[2], en el 31 de diciembre de 2010 habían 360 985 492 usuarios de internet y el 30 de enero de 2012 existían 2 405 518 376 usuarios de internet, el porcentaje de Penetración en el mundo fue de 34,3%, siendo Asia la región geográfica que posee la mayor cantidad de usuarios de internet (1076,7 millones) seguida de Europa (518,5 millones), Norte América (273,8 millones), América Latina / El Caribe (254,9 millones), África (167,3 millones), Medio Oriente (90 millones) y Oceanía/Australia (24,3 millones).

---

[1] "experto en informática, seguridad y redes de ordenadores", capaces de entrar en el ordenador más protegido de la compañía más importante, aunque para ello tenga que saltarse las medidas de seguridad más complejas.
[2] Unión Internacional de Telecomunicaciones (ITU), es el organismo especializado de las Naciones Unidas para las tecnologías de la información y la comunicación-TIC. http://www.itu.int/es/Pages/default.aspx

ESTADISTICAS MUNDIALES DEL INTERNET Y DE LA POBLACION						
Regiones	Poblacion (2012 Est.)	Usuarios Dic. 31, 2000	Usuarios Junio 30, 2012	% Población (Penetración)	Usuarios % Mundial	Facebook Sept 30, 2012
Africa	1,073,380,925	4,514,400	**167,335,676**	15.6 %	7.0 %	48,262,820
Asia	3,922,066,987	114,304,000	**1,076,681,059**	27.5 %	44.8 %	235,989,160
Europa	820,918,446	105,096,093	**518,512,109**	63.2 %	21.5 %	243,230,440
Oriente Medio	223,608,203	3,284,800	**90,000,455**	40.2 %	3.7 %	22,793,140
Norte America	348,280,154	108,096,800	**273,785,413**	78.6 %	11.4 %	184,177,220
Latinoamerica / Caribe	593,688,638	18,068,919	**254,915,745**	42.9 %	10.6 %	188,339,620
Oceania / Australia	35,903,569	7,620,480	**24,287,919**	67.6 %	1.0 %	14,614,780
TOTAL MUNDIAL	**7,017,846,922**	**360,985,492**	**2,405,518,376**	**34.3 %**	**100.0 %**	**937,407,180**

***Tabla 0.1 Estadística Mundial del Internet y de la Población** [2].*

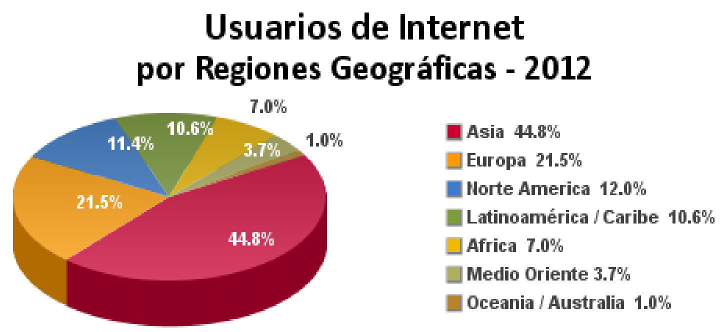


***Figura 0.1 Usuarios de Internet por Regiones Geográficas** [2].*

En marzo del 2013, una de las mayores firmas multinacionales de consultoría tecnológica "Forrester" anuncio que el comercio electrónico no frenará su crecimiento en Europa, y se va a convertir en el continente que liderará el mercado mundial de ecommerce en los próximos 4 años. La consultoría apuntaba que Europa alcanzará los 191 millones de euros en ventas en 2017, cifra claramente superior a los 113 millones actuales (CMT, tercer trimestre 2013).

Durante el tercer trimestre de 2012, las transacciones de comercio electrónico dentro de España supusieron un 43,1% del volumen de negocio total, mientras que las transacciones desde el exterior con España y desde España con el exterior han representado, respectivamente, un 15,1% y un 41,8% del volumen de negocio total (CMT, tercer trimestre 2013).

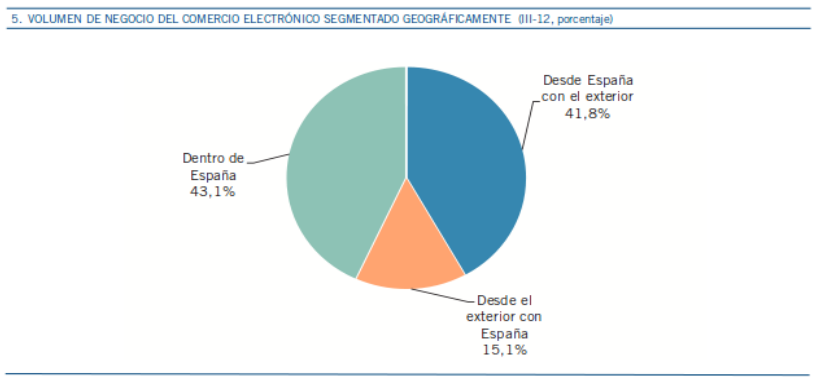


Figura 0.2 Volumen de negocio del comercio electrónico segmentado geográficamente (CMT, 2013).

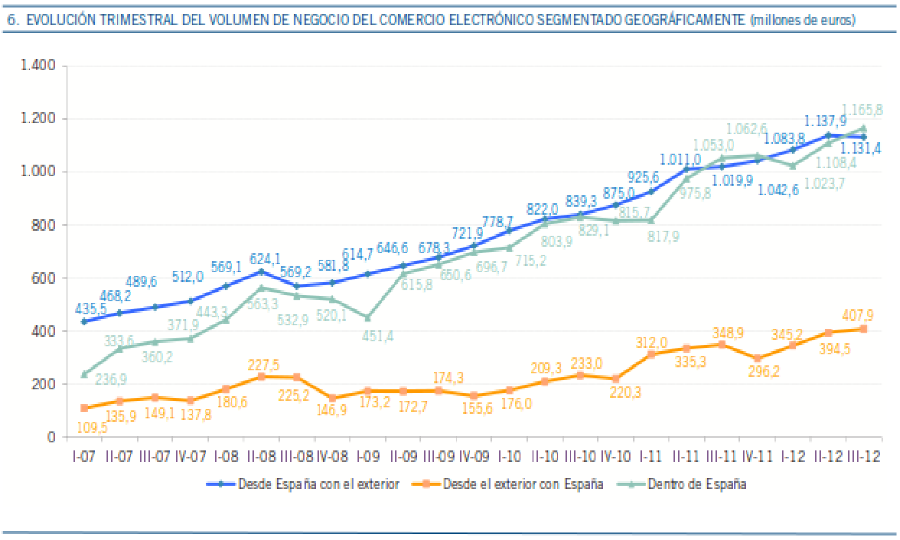


Figura 0.3 Evolución trimestral del Volumen de negocio del Comercio electrónico segmentado geográficamente (CMT, 2013).

A partir de febrero del 2012, ESET Latinoamérica notó un significativo aumento en la tasa de detección de un código malicioso un tanto particular: ACAD/Medre. Otra arista novedosa de este caso era que la mayoría de estas infecciones provenían principalmente de un país específico de América Latina, Perú. De acuerdo a información recopilada por el sistema de alerta temprana ESET Live Grid, el 96% de dichas detecciones provenían de esa nación.



Figura 0.4 Mapa de detección de ACAD/Medre (ESET, 2012).

Una investigación posterior permitió que este gusano diseñado para robar planos y proyectos realizados con AutoCad, logró sustraer aproximadamente 10 000 archivos pertenecientes a empresas del Perú. Por esta razón, la operación Medre se convirtió en el primer caso conocido de espionaje industrial en afectar la región de Latinoamérica. (ESET, 2012) La primera estadística que avala el vertiginoso aumento de malware para móviles, y que además, permite establecer como tendencia para el 2013 un crecimiento exponencial de códigos maliciosos para Android, es que en 2012 la cantidad de detecciones únicas crecieron 17 veces a nivel global con respecto a 2011. En relación a los países que experimentaron el mayor crecimiento de detecciones de malware para Android, expresado en cantidad de veces, se destacan Ucrania con 78 veces, Rusia 65, e Irán 48.

En cambio, si se considera la cantidad de detecciones ocurridas en 2012 independientemente de lo acontecido el año anterior, China, Rusia e Irán figuran entre los tres primeros con más cantidad de detecciones a nivel mundial. En sexto lugar aparece México, lo que lo posiciona como el país de Latinoamérica más afectado en cuanto a detecciones de códigos maliciosos para la plataforma móvil de Google. De acuerdo a estos antecedentes, se puede aseverar que el crecimiento de códigos maliciosos para Android en 2013 seguirá aumentando con mayor vertiginosidad.

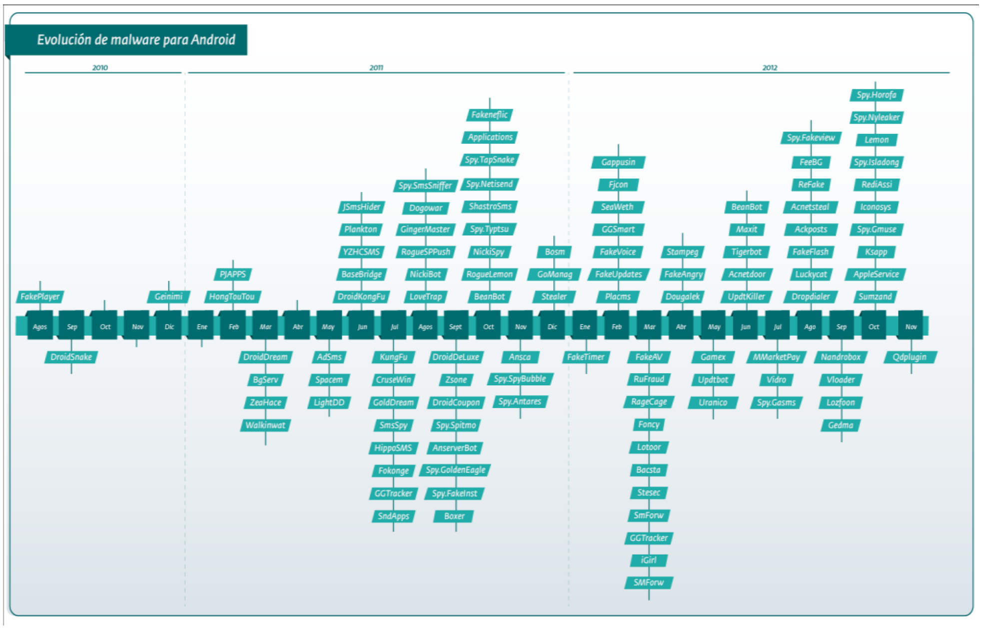


Figura 0.5 Surgimiento de familias de Malware para Android (ESET, 2012).

La expansión de la World Wide Web[3] ha proporcionado a los Blackhat Community[4] accesos ilimitados, a jugar y tender como presa a los administradores que carecen de conocimientos de la seguridad básica de una red. Ante esta situación debemos preguntarnos lo siguiente: ¿Qué es lo peor que podría pasar si una persona maliciosa tiene privilegios de acceso a los ordenadores de nuestra red, sin alguna autorización? ¿Podría este acontecimiento poner en aprietos o comprometer a la organización, a los clientes, o al país? ¿Cuánta productividad perdería la organización si todos sus discos duros fueran borrados?

Las redes de los ordenadores son consideradas como la espina dorsal de una organización. La información disponible es crucial para la organización y afectará en su capacidad de toma de decisiones. Es importante y crítica la tarea de asegurar las redes de la organización. Si el atacante es capaz de hacer esto, significará que la organización incurrirá en pérdidas significativas en términos de costo y datos confidenciales serán expuestos; existiendo casos en donde el compromiso de la información, ha causado la pérdida de vidas.

[3] World Wide Web (www), conocida también como la red informática mundial, es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de internet.
[4] Comunidad de sombreros negros, atacantes de élite, sus miembros poseen un perfil común en el que el elemento fundamental era su alto grado de preparación técnica y sus vastos conocimientos sobre el funcionamiento a bajo nivel de sistemas, redes y arquitecturas.

Al mismo tiempo las contramedidas se van haciendo más sofisticadas y extensamente usadas en la protección de datos e informaciones, como antivirus, antispam, firewall, IDS, IPS, balanceadores de cargas, control de accesos, sistemas criptográficos tipos VPN y entre otras. Todas estas contramedidas están basadas en la defensa, en la detección y el mecanismo de reacción. ¿Pero conocemos lo suficiente a nuestro enemigo? ¿O estos sistemas son siempre un paso por detrás de los Blackhat Community?. Como en lo militar, es muy importante el saber, quién es tu enemigo, qué tipo de estrategia utiliza, qué tipo de herramientas tiene y qué es lo que aspira?. Reunir todo este tipo de información no es tarea fácil, pero sí muy importante.

En la actualidad prestigiosas organizaciones de nivel mundial, llevan a cabo anonadados esfuerzos para la detección de los ciberataques en tiempo real, haciendo uso de sensores que actúan como "señuelos" para atraer ataques automatizados dirigidos a explotar las vulnerabilidades de los servicios de red, websites, smartphones y otros tipos de sistemas. Los incidentes detectados son mostrados en tiempo real y en mapas interactivos con diversas informaciones como son: su naturaleza, país de origen, servicios objetivos, estadísticas mensuales, etc.

A modo de ejemplo sirva la Figura 0.6, 0.7 y 0.8 para ilustrar la problemática de la seguridad en el mundo; según Deustche Telecom[5] , Global Security Map[6] y la comunidad Honeypot Project[7].

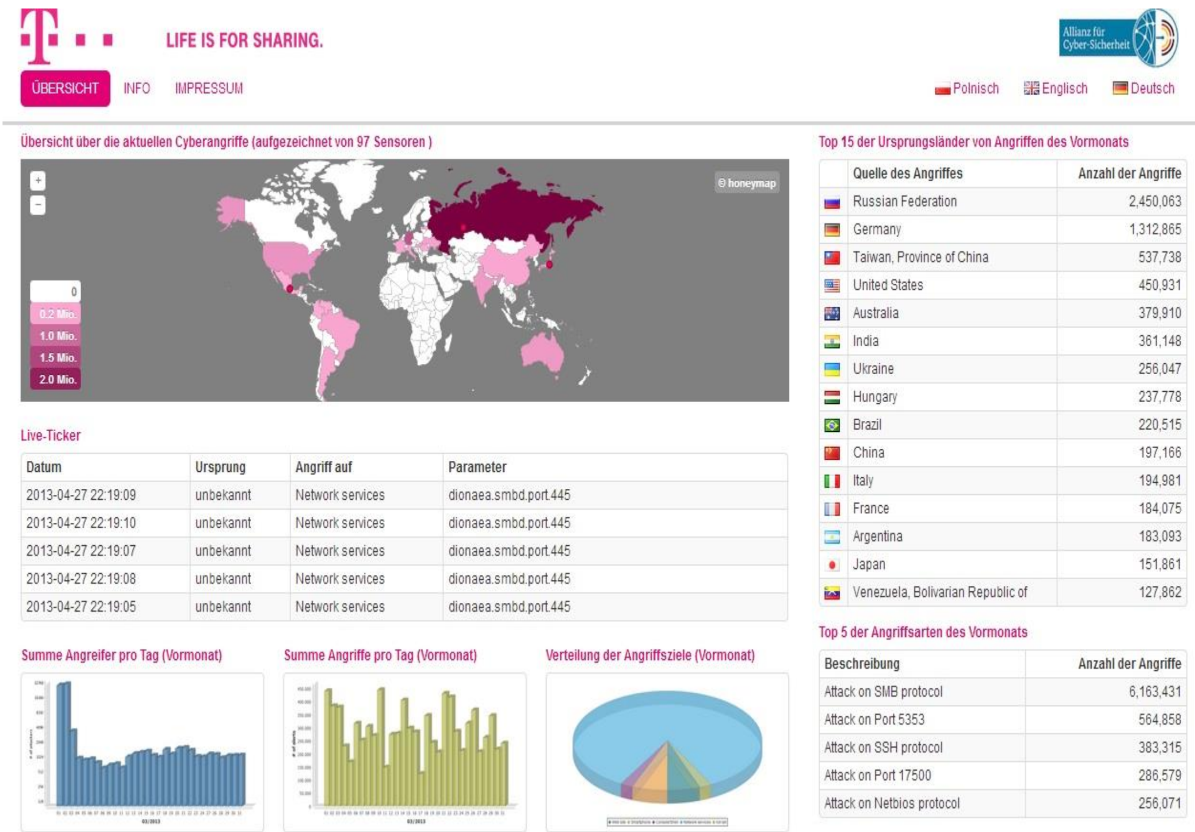


Figura 0.6 Ciberataques globales en tiempo real
***Deustche Telecom** [5]*

[5] Deustche Telecom, compañía de telecomunicaciones alemana desarrolló el proyecto Sicherheitstacho.eu, nombre que significa "medidor de seguridad", proyecto que recopila los datos detectados por 97 sensores distribuidos por todo el mundo haciendo uso de los honeypots en tiempo real. http://www.sicherheitstacho.eu/

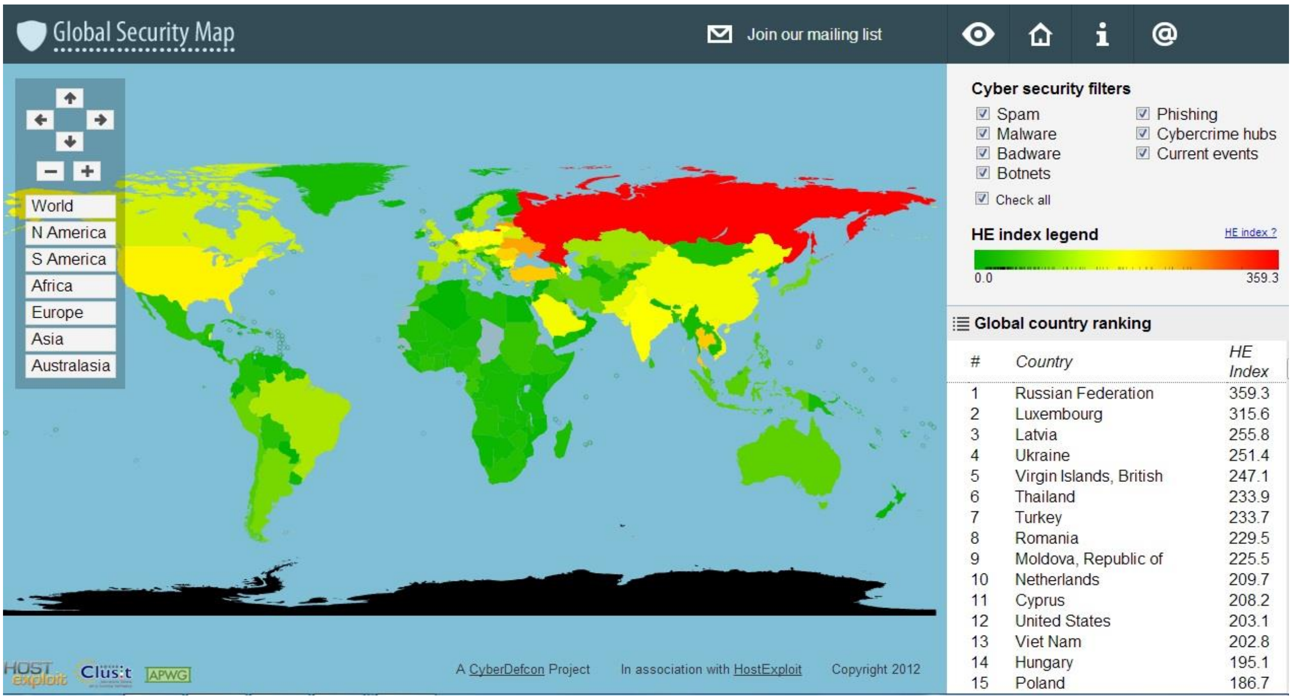


Figura 0.7 Ciberataques globales en tiempo real
***Global Security Map** [6].*

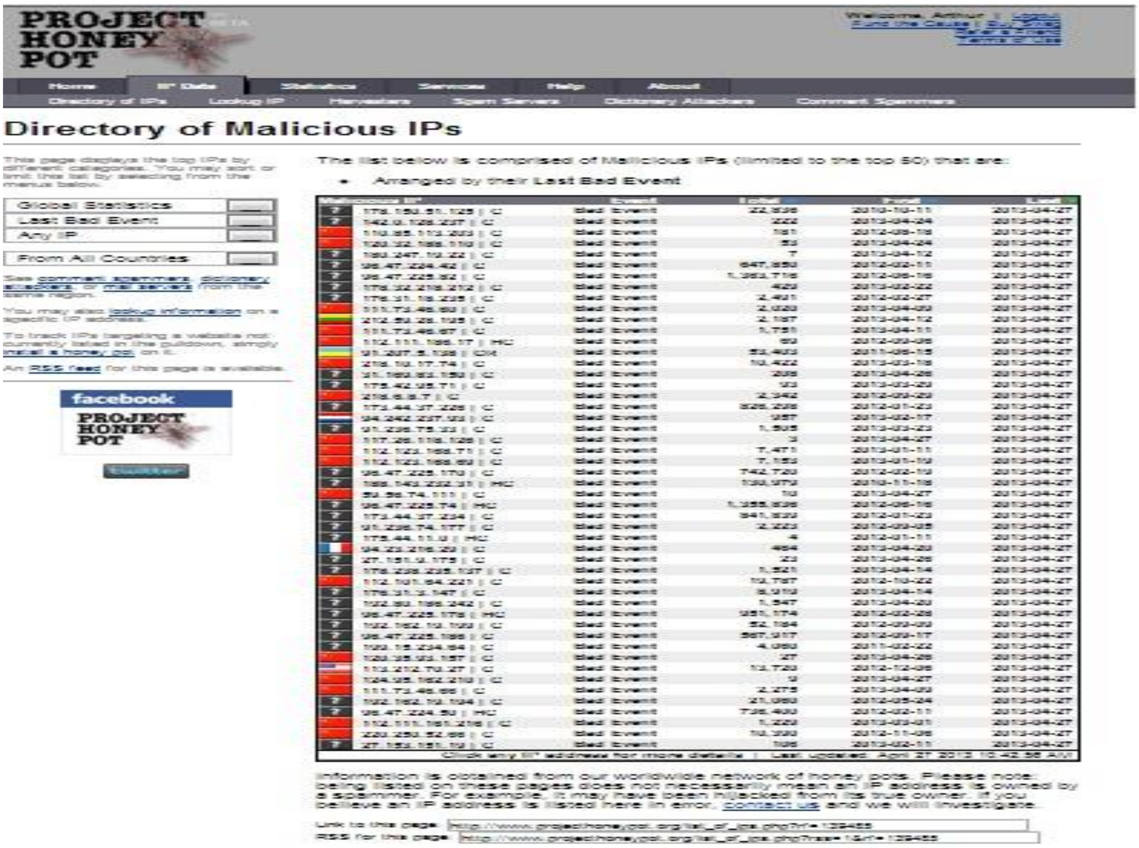


Figura 0.8 Directorio de direcciones IP de los Servidores de Spam
***Project Honey Pot** [7].*

[6] Global Security Map, proyecto desarrollado por la organización de voluntarios HostExploit con colaboración del grupo Ruso Group-IB y CSIS en Dinamarca. http://globalsecuritymap.com/
[7]Project Honey Pot, fue creado por Unspam Technologies, In, compañía antispam. https://www.projecthoneypot.org/about_us.php

Los sistemas de detección de intrusos son herramientas que han sido diseñadas para aumentar la seguridad de una red de datos. Los estudios realizados en esta área tuvieron su origen en el análisis de los flujos de datos de los sistemas vulnerados; inicialmente personal experto en el tema, dedicó todo su tiempo a identificar el origen de las anomalías y evitar que afectara de nuevo al sistema de información, esto evolucionó en lo que se conoce como sistemas de detección basados en reglas, que han sido el estándar hasta la actualidad.

Los sistemas basados en reglas requieren actualización permanente y dándose el caso en que un ataque es modificado levemente, el sistema es incapaz de detectarlo. Por esto la inteligencia artificial se ha planteado como una solución a las limitaciones de los sistemas tradicionales.

MOTIVACION PARA LA REALIZACIÓN DEL PROYECTO DE FIN DE MÁSTER

La motivación que me llevó a la elaboración del presente proyecto fin de máster fueron las siguientes razones:

a) La mayoría de los sistemas computacionales utilizados poseen alguna falla de seguridad que puede ser explotada por usuarios mal intencionados. Investigar y corregir todas las potenciales fallas de seguridad no es viable técnica y comercialmente.

b) Inviabilidad de la sustitución de sistemas con fallas de seguridad, por versiones más seguras.

c) Desarrollo de sistemas completamente libres de fallas de seguridad. ”Cabe aclarar que es una tarea virtualmente imposible”.

d) El abuso de privilegios de los usuarios internos (motivados por la competencia, problemas personales, el descontento con la organización, etc.) pueden comprometer un sistema altamente seguro.

OBJETIVOS DEL PROYECTO DE FIN DE MÁSTER

El presente PFM se centrará principalmente en el enfoque de detección de anomalías centrándose en la capacidad de generalización de algunas herramientas basadas en aprendizaje no supervisado, estudiando la posibilidad de detectar también nuevos ataques (Zero day attack), teniendo como objetivos principales en el desarrollo de la presente memoria lo siguiente:

- Diseño de un sistema capaz de detectar situaciones anómalas en la red, basadas en modelos de inteligencia computacional.

- Instrumento complementario para visualizar la estructura interna de un conjunto de datos asociado al tráfico de redes.

ORGANIZACIÓN DEL PROYECTO DE FIN DE MÁSTER

El presente PFM ha sido organizado de la siguiente manera: El Capítulo Primero, se definirán algunos conceptos fundamentales de Seguridad, Seguridad de la

Información, conceptos básicos de las técnicas de ataque, las técnicas de protección y detección de intrusos. En el Capítulo Segundo, se muestran conceptos teóricos relacionados a los Sistemas de Detección de Intrusos (IDS), sus limitaciones, una breve definición de los Honeypots y las técnicas de visualización para la seguridad en la red. En el Capítulo Tercero, hablaremos de las redes neuronales artificiales (ANNs), su inspiración biológica, el aprendizaje automático y los algoritmos de aprendizaje. En el Capítulo Cuarto se abordarán los detalles de la solución propuesta para la detección de intrusos con redes neuronales (para redes TCP/IP), para mejorar el desempeño de la red y posibilitar la aplicación futura de este método en la detección en tiempo real de ataques en redes de computadores, se hará mención de aquellos trabajos y artículos científicos relevantes en la ID relacionados con la propuesta, se detallaran temas como: la aplicación del modelo neuronal no supervisado para la proyección del tráfico de datos, ataques de SNMP, las herramientas utilizadas para las simulaciones de las redes neuronales, las etapas de implementación del proyecto, la captura, selección y segmentación de los datos, el pre-procesado y análisis de los datos. En el Capítulo Quinto, se llevarán a cabo los experimentos y la visualización de los resultados; y por último en el Capítulo Sexto, se presentarán las conclusiones y los trabajos futuros a raíz de los estudios llevados a cabo durante el desarrollo del presente proyecto de fin de máster.

1. CAPÍTULO PRIMERO

"Esto es lo que llamamos Criptograma, en el cual el sentido está oculto bajo letras embarulladas a propósito y que, convenientemente dispuestas, formarían una frase inteligible - dijo el profesor -

Viaje al centro de la tierra. Julio Verne

1.1 CONCEPTOS FUNDAMENTALES

Es importante la presentación de algunos conceptos básicos, sobre los cuales está basada el presente proyecto de fin de Máster. En primer lugar se definirán algunos conceptos fundamentales de Seguridad, Seguridad de la Información y a continuación se pasará a definir conceptos básicos de técnicas de ataque, técnicas de protección y detección de intrusos.

1.1.1 LA SEGURIDAD

En el lenguaje de las ciencias sociales, "seguridad" es un concepto controvertido, ya que no existe un consenso generalizado sobre su significado. En función de las personas, sus ideas, cultura y percepciones de la realidad el término adquiere un valor distinto.

Este hecho se ve corroborado por el importante número de definiciones de seguridad que han aparecido, sobre todo a partir del final de la Guerra Fría. Seguridad Nacional, Seguridad Común, Seguridad Colectiva, Seguridad Compartida, Seguridad Humana o Seguridad Cooperativa efectúan una descripción de lo que sus ideólogos consideran que debe ser entendido por seguridad y, lo que es quizás más importante, como conseguirla. En los últimos años, a los términos ya citados se les han ido uniendo en los últimos años de seguridad sostenible, así como conceptos híbridos tales como "poder duro-poder blando"[8] o "poder inteligente"[9] (IEEE, Junio 2011).

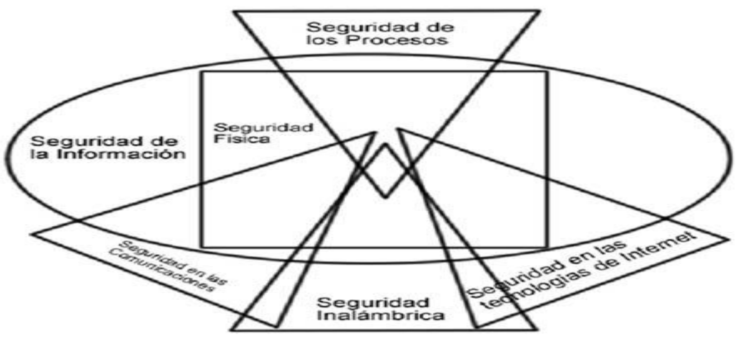


Figura 1.1 Mapa de Seguridad (IEEE, Junio 2011)

[8] Poder Duro - Poder Blando, **Hard power** (*Poder duro*) es un concepto que es principalmente usado en las relaciones internacionales, y que se refiere al poder nacional radicado en los medios militares y económicos. Es usado en contraste con el término *Soft power* (*Poder blando*), que se refiere al poder con origen en la diplomacia, la cultura y la historia.
[9] Poder Inteligente, **Smart power** (Poder inteligente) es un término en las relaciones internacionales definido por Joseph Nye como "La capacidad de combinar **Hard** y **Sof power**" en una estrategia vencedora.

1.1.2 LA SEGURIDAD DE LA INFORMACIÓN

Según todos los expertos y por lo expuesto en la presente memoria coincidimos que la seguridad total no existe, así pues no se habla de máxima seguridad sino de mínima inseguridad, detallaré a continuación los conceptos más relevantes que en la actualidad tienen influencia en el mundo académico con relación a la seguridad de la información:

"Es la capacidad de las Redes o de los sistemas de información de resistir, con un determinado nivel de confianza, todos los accidentes o acciones ilegales o malintencionadas que pongan en peligro la disponibilidad, autenticidad, integridad y confidencialidad de los datos almacenados y transmitidos; y de los correspondientes servicios que dichas redes y sistemas ofrecen a hacen accesibles"**[10]**.(ENISA,2002)

"Preservación de la confidencialidad, integridad y disponibilidad de la información; además, también pueden estar involucradas otras propiedades como la autenticidad, responsabilidad, no-repudio y confiabilidad".**[11]** (ISO/IEC 17799:2005)

"Information is an asset to all individuals and businesses. Information Security refers to the protection of these assets in order to achieve C - I - A as the following diagram:
Confidentiality *- protecting information from being disclosed to unauthorised parties.*
Integrity *- protecting information from being changed by unauthorised parties.*
Availability *- to the availability of information to authorised parties only when requested."****[12]***



Figura 1.2 Information Security- Infosec **[12].**

"La seguridad de la información es el conjunto de medidas preventivas y reactivas de las organizaciones y de los sistemas tecnológicos que permitan resguardar y proteger la información buscando mantener la confidencialidad, la disponibilidad e integridad de la misma."**[13]**

---

[10] Concepto de la Agencia Europea de Seguridad de las Redes y la Información (ENISA), cuya misión es la de ayudar a la Comunidad a obtener unos niveles particularmente altos de seguridad de las redes y de la información.
[11] ISO (Organización Internacional de Normalización), es el organismo encargado de promover el desarrollo de normas internacionales de fabricación tanto de productos como de servicios, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica.
[12] Infosec (Information Security), entidad del gobierno de Hong kong que concientiza al sector público sobre la seguridad de la información así como medidas para la prevención de crimenes ciberneticos. http://www.infosec.gov.hk
[13] http://es.wikipedia.org/wiki/Seguridad_de_la_informaci%C3%B3n

“Ensures that within the enterprise, information is protected against disclosure to unauthorized users (confidentiality), improper modification (integrity) and non - access when required (availability).”**[14]** (ISACA, November 2012)

En la seguridad de la información es importante señalar que su manejo está basado en la tecnología y debemos de saber que puede ser confidencial: la información está centralizada y puede tener un alto valor. Puede ser divulgada, mal utilizada, ser robada, borrada o saboteada. Esto afecta su disponibilidad y la pone en riesgo. La información es poder, y según las posibilidades estratégicas que ofrece tener acceso a cierta información, ésta se clasifica como:

Crítica: Es indispensable para la operación de la empresa.
Valiosa: Es un activo de la empresa y muy valioso.
Sensible: Debe de ser conocida por las personas autorizadas.

Existen dos palabras muy importantes que son riesgo y seguridad:

Riesgo: Es todo tipo de vulnerabilidades, amenazas que pueden ocurrir sin previo aviso y producir numerosas pérdidas para las empresas. Los riesgos más perjudiciales son a las tecnologías de información y comunicaciones.
Seguridad: Es una forma de protección contra los riesgos.

La seguridad de la información comprende diversos aspectos entre ellos la disponibilidad, comunicación, identificación de problemas, análisis de riesgos, la integridad, confidencialidad, recuperación de los riesgos. Como hemos podido observar la definición de seguridad de la información atiende los siguientes requisitos:

- Confidencialidad, Es la propiedad de prevenir la divulgación de información a personas o sistemas no autorizados. A groso modo, la confidencialidad es el acceso a la información únicamente por personas que cuenten con la debida autorización.
- Integridad, Es la propiedad que busca mantener los datos libres de modificaciones no autorizadas. (No es igual a integridad referencial en bases de datos.) A groso modo, la integridad es el mantener con exactitud la información tal cual fue generada, sin ser manipulada o alterada por personas o procesos no autorizados.
- Disponibilidad, La disponibilidad es la característica, cualidad o condición de la información de encontrarse a disposición de quienes deben acceder a ella, ya sean personas, procesos o aplicaciones. Groso modo, la disponibilidad es el acceso a la información y a los sistemas por personas autorizadas en el momento que así lo requieran.
- Autenticación o autentificación, Es la propiedad que me permite identificar el generador de la información. Por ejemplo al recibir un mensaje de alguien, estar seguro que es de ese alguien el que lo ha mandado, y no una tercera persona haciéndose pasar por la otra (suplantación de identidad).

[14] “Asegura que dentro de la empresa, la información es protegida contra la revelación de usuarios no autorizados (la confidencialidad), la modificación impropia (la integridad) y el no acceso cuando es requerido (la disponibilidad)”

En un sistema informático se suele conseguir este factor con el uso de cuentas de usuario y contraseñas de acceso.

- No repudio, Proporciona protección contra la interrupción, por parte de alguna de las entidades implicadas en la comunicación, de haber participado en toda o parte de la comunicación. El servicio de Seguridad de No repudio o irrenunciabilidad está estandarizado en la ISO-7498-2.**[15]**

1.1.3 TÉCNICAS DE ATAQUE

Los ataques a los computadores se dan por la exploración de vulnerabilidades existentes en sus sistemas. Un punto débil en la estructura de un recurso computacional se caracteriza como Vulnerabilidad. (MARTÍN, 2006)

> *"Un ataque puede ser definido como una acción maliciosa que viola las políticas de seguridad y compromete la integridad o disponibilidad de los recursos en un sistema. Posee como principio elemental la exploración de vulnerabilidades, tanto de sistemas como de protocolos existentes. Los resultados de un ataque pueden caracterizarse como una invasión o ocasionar indisponibilidad de los servicios suministrados". (MACHADO, 2005)*

Las amenazas de las informaciones pueden ser encuadradas en las clases de violaciones visualizadas en la Figura 1.3 que son descritas de la siguiente manera (DE LIMA, 2005; MACHADO, 2005):

- Interrupción: La transmisión de las informaciones es interrumpida, impidiendo que la misma llegue a su destino.
- Interceptación: Las informaciones son interceptadas durante la transmisión, caracterizando así un ataque a la confidencialidad.
- Modificación: Las informaciones son interceptadas, alteradas y reenviadas al destinatario. Ese tipo de ataque compromete la confidencialidad e integridad de la información
- Fabricación: se caracteriza por la inserción de informaciones en los sistemas.

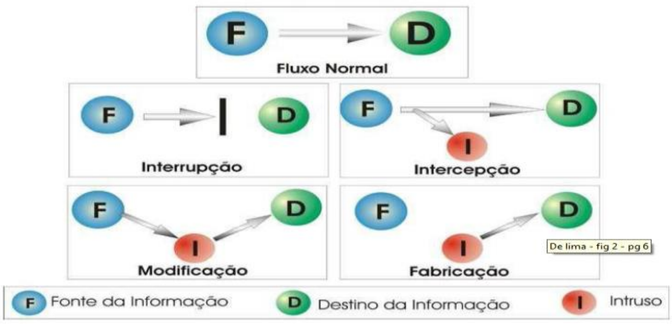


Figura 1.3 Clases de violaciones a la información (De Lima, 2005)

[15] ISO-7498-2, estándar en la que se ha definido una arquitectura general de normas de seguridad.

La evolución progresiva de ataques podría ser principalmente atribuida a la naturaleza dinámica de los sistemas y redes, la creatividad de los atacantes y la amplia gama computacional de hardware y software. Tal complejidad surge cuando tratando con sistemas distribuidos en base a red y redes inseguras como el Internet. Una representación gráfica de las técnicas aplicadas por el atacante en base a sus conocimientos y la sofisticación de los ataques la podemos ver en la Figura 1.4.

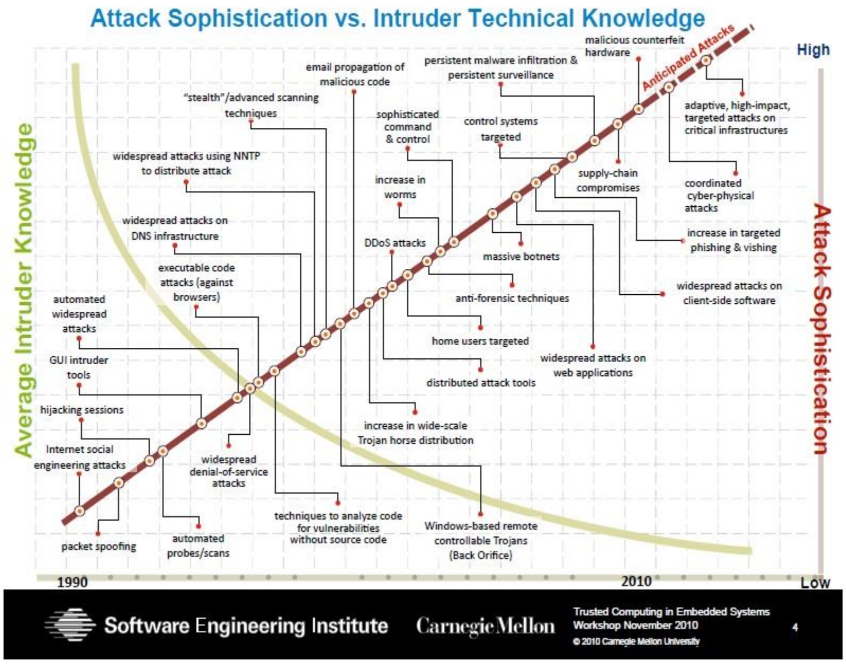


Figura 1.4 Evolución de la sofisticación de ataques/ Niveles de conocimientos de intrusión (TURBYFILL, 2011)

> *there is an inverse relationship between the decreasing technical knowledge required to execute attacks and the increasing sophistication of those attacks. In other words, "less skill is needed to do more damage".*
>
> *Any attack on or intrusion into a system is likely to affect at least one of the three principles of information security: integrity, confidentiality, and availability of information. Thus, an intrusion can be generally defined as "any set of actions that attempt to compromise the integrity, confidentiality or availability of a re-source" (CORCHADO, HERRERO, 2013)*

Un atacante generalmente utiliza una herramienta para explorar vulnerabilidades y así realizar la acción deseada en su objetivo. Esas herramientas son conocidas como exploits.

Las acciones realizadas por los atacantes, generalmente siguen las etapas ilustradas en la Figura 1.5 (BASTOS, 2004; DE LIMA, 2005; WEISSHEIMER,2006).

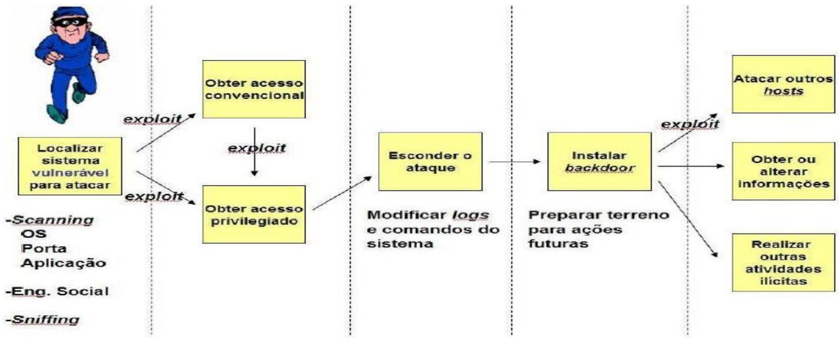


Figura 1.5 Etapas de ataques típicos
(BASTOS, 2004; DE LIMA, 2005; WEISSHEIMER, 2006)

Los ataques normalmente buscan invadir el sistema o indisponer los servicios ofrecidos por el sistema. La forma como el atacante actúa sobre el sistema, autores como (De Lima, 2005; Machado, 2005 y Marcelo, 2003) lo clasifican de la siguiente manera:

1.1.3.1 ATAQUES DE EXPLORACIÓN

Es el tipo de ataque que tiene como objetivo extraer el máximo de informaciones sobre la víctima. Dentro de las informaciones que pueden ser coleccionadas son: nombres y direcciones de servidores, servicios activos, sistemas operacionales instalados, versiones de los software y servicios utilizados, entre otras.

Dentro de este tipo de ataque podemos clasificar las siguientes herramientas:

- Sniffers: Herramienta que analiza el tráfico de la red, dándonos la posibilidad de la captura de paquetes de datos y la verificación del contenido transportado.
- Port Scanners: Herramienta que obtiene informaciones de la red, tales como cuantos computadores están activos y que servicios están configurados en la red, a través del análisis de las puertas de servicio TCP y UDP **[16]**.

1.1.3.2 ATAQUES DE DENEGACIÓN DE SERVICIO (DENIAL OF SERVICE-DOS)

Es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales del sistema de la víctima. Un ejemplo notable de este tipo de ataque se produjo el 27 de marzo del 2013, cuando un ataque de una empresa a otra inundó la red de spam provocando una ralentización generalizada de Internet e incluso llegó a afectar a puntos clave como el nodo central de Londres **[17]**.

---

[16] TCP y UDP, protocolos utilizados por el modelo TCP/IP, precisamente en la capa de transporte. El primero orientado a conexión y el otro sin conexión.
[17] La holandesa Cyberbunker lanza un ataque masivo de spam y denegación de servicio contra Spamhaus, firma suiza de seguridad. http://tecnologia.elpais.com/tecnologia/2013/03/27/actualidad/1364405367_942632.html

Se genera mediante la saturación de los puertos con flujo de información, haciendo que el servidor se sobrecargue y no pueda seguir prestando servicios; por eso se le denomina “denegación, pues hace que el servidor no de abasto a la cantidad de solicitudes. Esta técnica es usada por los llamados crackers **[18]** para dejar fuera de servicio a servidores objetivo.

Para el ataque DoS existe una evolución de ataque que sería el ataque de denegación de servicio distribuido (Distributed Denial of Service-DDoS) que es el más temido y mucho más destructivo (MARCELO, 2003). El DDoS posee el mismo principio de funcionamiento del DoS, pero su origen de ataque es distribuido; osea, parte de diversos computadores al mismo tiempo (STEFFEN, 2003). Para realizar ese tipo de ataque, el atacante obtiene, generalmente a través de invasión, acceso a varios computadores (conocidos por daemons**[19]**), donde instala y ejecuta un programa que irá lanzar ataques DoS. Ese tipo de ataque es ilustrado en la Figura 1.6.

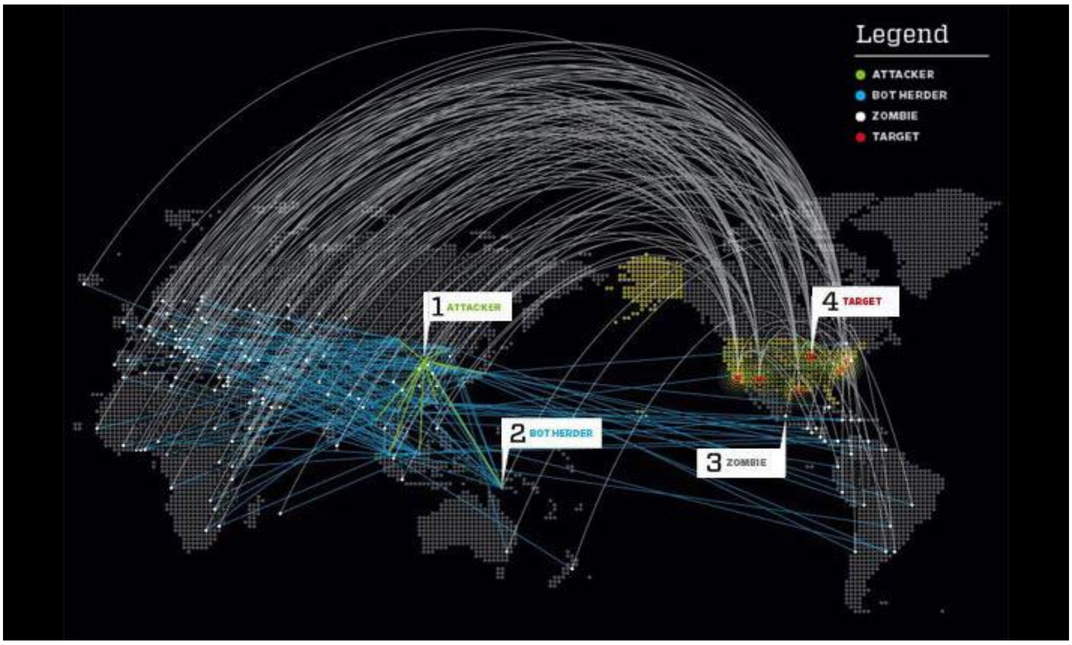


Figura 1.6 Ataque de DoS (Denial of Service).
Fuente: http://datavisualization.ch/showcases/how-twitter-got-attacked-by-a-ddos/

1.1.3.3 ATAQUES DE PENETRACIÓN

Este tipo de ataque tiene como objetivo obtener control de un computador a través de la exploración de las fallas de un programa. La falla que es explorada en ese ataque es el desbordamiento de buffer, del inglés buffer overflow.

[18] El término cracker se utiliza para referirse a las personas que rompen algún sistema de seguridad. Los crackers pueden estar motivados por una multitud de razones, incluyendo fines de lucro, protesta, o por el desafío.
[19] Un demonio, daemon (nomenclatura usada en sistemas UNIX y UNIX-like), servicio (nomenclatura usada en Windows) o programa residente (nomenclatura usada en Ms-Dos) es un tipo especial de proceso informático no interactivo, es decir, que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario. Este tipo de programas continua en el sistema, es decir, que puede ser ejecutado en forma persistente o reiniciado si se intenta matar el proceso dependiendo de configuración del demonio y políticas del sistema. La palabra demonio viene de las siglas en ingles D.A.E.M.O.N (Disk And Execution Monitor).

"Es un tipo de vulnerabilidad que afecta al software y es muy utilizada para realizar ataques dirigidos a conseguir que el programa realice las acciones que el atacante, y no el propio programa, quiera. Son defectos en la programación que provocan un error o el cuelgue del sistema pero son provocados de forma intencionada."(INTECO, desbordamiento de buffer)

Si hiciéramos una comparación, el desbordamiento de buffer provoca algo similar a lo que ocurre cuando llenamos un vaso más allá de su capacidad: éste se desborda y el contenido se derrama. Cuando el programador no incluye las medidas necesarias para comprobar el tamaño del buffer en relación con el volumen de datos que tiene que alojar, se produce también el derramamiento de estos datos que se sobrescriben en otros puntos de la memoria, lo cual puede hacer que el programa de errores o incluso se cuelgue.

El atacante calcula qué cantidad de datos necesita enviar para conseguir saber cuándo se producirá el desbordamiento y dónde se reescribirán los datos y posteriormente consigue el desbordamiento, en definitiva, que el programa ejecute el código que él ha enviado. Este tipo de vulnerabilidad, dado que se produce por un defecto en el código del programa, sólo puede ser solventada mediante las actualizaciones o parches del programa en cuestión, lo cual hace muy necesario mantener actualizados todos los programas instalados en nuestro equipo.

Uno de los ejemplos más famosos de ataque mediante este sistema, fue el del gusano BLASTER **[20]**, que aprovechaba una debilidad del sistema operativo Windows, para introducirse en los ordenadores directamente desde Internet a través del puerto de comunicaciones 135. Tras ello, provocaba un desbordamiento de buffer en la máquina afectada. (INTECO, desbordamiento de buffer)

1.1.3.4 INGENIERIA SOCIAL

La ingeniería social es una forma de obtener informaciones o acceso a sistemas contando con la ingenuidad, inocencia y confianza de un ser humano.

"El termino ingeniería social está siendo utilizado para definir técnicas utilizadas para sacar provecho de fallas que las personas cometen o que sean llevadas a cometer con relación a las informaciones del área de tecnología de la información"

"La ingeniería social es la habilidad de un hacker manipular la tendencia humana natural de confianza con el objetivo de obtener informaciones a través de un acceso válido en un sistema no autorizado. El propósito básico de la ingeniería social es prácticamente el mismo de los hackers en general: obtener acceso no autorizado a sistemas o informaciones para cometer fraudes, invadir redes, espionaje industrial, robar identidades o simplemente interrumpir sistemas o redes". (KLEIN, 2005)

[20] Blaster, (o también llamado Lovsan o LoveSan 3a1) es un gusano de red de Windows que se aprovecha de una vulnerabilidad en el servicio DCOM para infectar a otros sistemas de forma automática. El gusano fue detectado y liberado el día 11 de agosto de 2003. La tasa de infecciones aumentó considerablemente hasta el día 13 de agosto de 2003. Gracias al filtrado por las ISP's y la gran publicidad frente este gusano, la infección pudo frenarse. El 29 de agosto de 2003, Jeffrey Lee Parson, de 18 años de Hopkins, Minnesota fue arrestado por crear la variante B del gusano Blaster; admitió ser el responsable y fue sentenciado a 18 meses en prisión en enero de 2005.

A través de las técnicas de ingeniería social es posible obtener informaciones de una empresa de forma simple, el atacante se hace pasar por otra persona y solicita informaciones a un funcionario a través de una llamada telefónica; y por medio de la persuasión lograr que la persona brinde las informaciones requeridas por el atacante. Ejemplos relacionados a ataques de ingeniería social lo podemos encontrar en (INTECO-CERT, noticias en la red).

1.1.4 TÉCNICAS DE PROTECCIÓN

Una vez conocidas las vulnerabilidades y ataques a las que está expuesto un sistema es necesario conocer los recursos disponibles para protegerlo. Mientras algunas técnicas son evidentes (seguridad física por ejemplo) otras pautas no lo son tanto e incluso algunas pueden ocasionar una sensación de falsa seguridad.

Siendo reiterativo, ninguna de las técnicas expuestas a continuación representarán el 100% de la seguridad deseado, aunque muchas parezcan la panacea, será la suma de algunas de ellas las que convertirán un sistema interconectado en confiable.

1.1.4.1 VULNERAR PARA PROTEGER

Los intrusos utilizan diversas técnicas para quebrar los sistemas de seguridad de una red. Básicamente buscan los puntos débiles del sistema para poder colarse. El trabajo de los administradores y Testers no difiere mucho de esto. En lo que sí se diferencian y por completo es en los objetivos: mientras que un intruso penetra en las redes para distintos fines (investigación, daño, robo, etc) un administrador lo hace para poder mejorar los sistemas de seguridad. Al conjunto de técnicas que se utilizan para evaluar y probar la seguridad de una red se conoce como Penetration Testing[21], uno de los recursos más poderosos con los que se cuenta hoy para generar barreras cada vez más eficaces.

El software y hardware utilizados son una parte importante pero no la única. A esto se agregan las Políticas de seguridad internas, que deben ser generadas e implementadas por la organización.

1.1.4.2 ADMINISTRACION DE LA SEGURIDAD

Podemos dividirla en tres bloques:

- Autenticación: Se refiere a establecer las entidades que pueden tener acceso al universo de recursos de cómputo que cierto medio ambiente puede ofrecer.

[21]Penetration Testing – ISACA, definition: A live test of the effectiveness of security defenses through mimicking the actions of real-life attackers. http://www.isaca.org/Pages/Glossary.aspx?tid=651

- Autorización: es el hecho de que las entidades que pueden tener acceso al universo de recursos de cómputo, tengan acceso únicamente a las áreas de trabajo sobre las cuales ellas deben tener dominio.
- Auditoria: es la continua vigilancia de los servicios de producción. Entra dentro de este grupo el mantener estadísticas de acceso, estadísticas de uso y políticas de acceso físico a los recursos.

Política	Procedimiento	Práctica
Necesitamos proteger nuestro servidor Web contra accesos no autorizados.	Se mantendrá actualizado el servidor Web en cuanto a seguridad.	Se comprobará diariamente si existen parches de seguridad del servidor Web, en cuyo caso se aplicarán.
	Se instalará un IDS configurado para comprobar que la actividad en el servidor Web es normal.	Se instalará la última versión del "Snort", y se aplicarán los cambios de configuración pertinentes para concentrar la vigilancia especialmente en el servidor Web.

Tabla 1.1 Ejemplo de política de seguridad (BORGHELLO, 2001).

Como primer paso una organización debe disponer de políticas, puesto que reflejan la “voluntad de hacer algo” que permita detener un posible ataque antes de que este se lleve a cabo (proactividad). Entre los métodos de protección más comunes tenemos:

- Sistemas orientados a conexión de red

  Monitorizan las conexiones que se intentan establecer en una red o equipo en particular, siendo capaces de efectuar una acción sobre la base de métricas como: origen y destino de la conexión, servicio solicitado, permisos, etc. Las acciones que pueden emprender suelen ir desde rechazo de la conexión hasta alerta al administrador. En esta categoría están los cortafuegos (Firewalls)**[23]** y los Wrappers.**[24]**

- Sistemas de análisis de vulnerabilidades

  Analizan sistemas en busca de vulnerabilidades conocidas anticipadamente. La “desventaja” de estos sistemas es que pueden ser utilizados tanto por personas autorizadas como por personas que buscan acceso no autorizado al sistema.

[22]Definición dada por Khaled Labib en su obra “Computer Security and Intrusion Detection.”(KHALED, 2004)
[23] Un cortafuegos (firewall en inglés) es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas. Los cortafuegos pueden ser implementados en hardware o software, o una combinación de ambos.
[24] TCP Wrapper ("Envoltorio de TCP") es un sistema de red ACL que trabaja en terminales y que se usa para filtrar el acceso de red a servicios de protocolos de Internet que corren en sistemas operativos (tipo UNIX), como Linux o BSD. Permite que las direcciones IP, los nombres de terminales y/o respuestas de consultas ident de las terminales o subredes sean usadas como tokens sobre los cuales filtrar para propósitos de control de acceso.

- Sistemas de protección a la integridad de información

  Sistemas que mediante criptografía o sumas de verificación tratan de asegurar que no ha habido alteraciones indeseadas en la información que se intenta proteger. Algunos ejemplos son los programas que implementan algoritmos como Message Digest (MD5)**[25]** o Secure Hash Algorithm (SHA)**[26]**, o bien sistemas que utilizan varios de ellos como PGP**)[27]**, Tripwire**)[28]** y dozeCrypt**[29]**

- Sistemas de detección de intrusos (IDS)

  "An IDS can be defined as a piece of software that runs on a host, which monitors the activities of users and programs on the same host and/or the traffic on networks to which that host is connected **[22]**. The main purpose of an IDS is to alert the system administrator to any suspicious and possibly intrusive event taking place in the system that is being analysed. Thus, they are designed to monitor and to analyse computer and/or network events in order to detect suspect patterns that may relate to a system or network intrusion." (HERRERO, CORCHADO, 2013). Más detalles sobre IDS serán vistos en el capítulo segundo.

- Sistemas de protección a la privacidad de la información

  Herramientas que utilizan criptografía para asegurar que la información solo sea visible para quien tiene autorización. Su aplicación se realiza principalmente en las comunicaciones entre dos entidades. Dentro de este tipo de herramientas se pueden citar a Pretty Good Privacy(PGP**)[27]**, Secure Sockets Layer(SSL)**[28]** y los Certificados digitales**[29]**

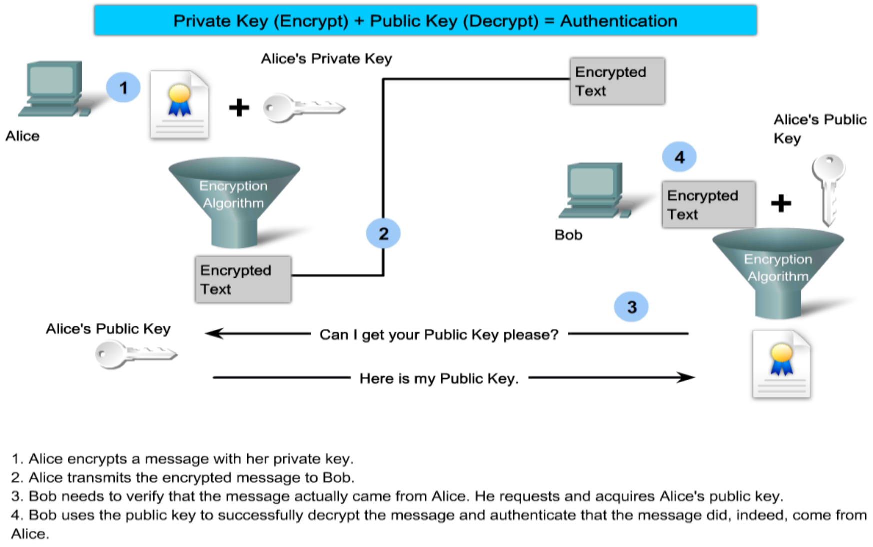


Figura 1.7 Certificado de autenticación.
Fuente: http://www.diazantuna.es/certificados-digitales/

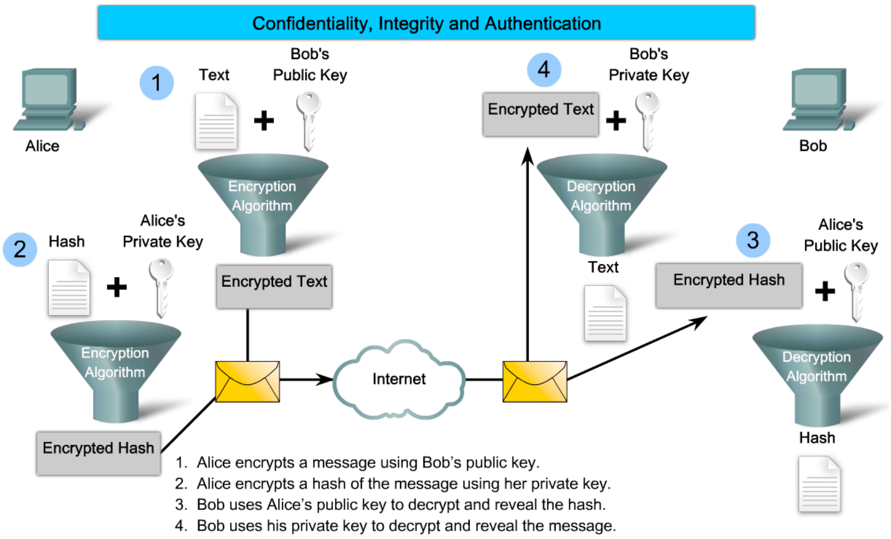


Figura 1.8 Certificado de cifrado.
Fuente: **http://www.diazantuna.es/certificados-digitales/**

Resumiendo, un modelo de seguridad debe estar formado por múltiples componentes o capas que pueden ser incorporadas de manera progresiva al modelo global de seguridad en la organización, logrando así el método más efectivo para disuadir el uso no autorizado de sistemas y servicios de red. Podemos considerar que estas capas son:

1. Política de seguridad de la organización.
2. Auditoria.
3. Sistemas de seguridad a nivel de Router-Firewall.
4. Sistemas de detección de intrusos.
5. Plan de respuesta a incidentes.
6. Penetration testing[21].

[21]Penetration Testing – ISACA, definition: A live test of the effectiveness of security defenses through mimicking the actions of real-life attackers. http://www.isaca.org/Pages/Glossary.aspx?tid=651
[25] En criptografía, MD5 (abreviatura de Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado. Diseñado por el profesor Ronald Rivest del MIT (Massachusetts Institute of Technology, Instituto Tecnológico de Massachusetts). Fue desarrollado en 1991 como reemplazo del algoritmo MD4 después de que Hans Dobbertin descubriese su debilidad.
[26] La familia SHA (Secure Hash Algorithm, Algoritmo de Hash Seguro) es un sistema de funciones hash criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos y publicadas por el National Institute of Standards and Technology (NIST).
[27] PGP es un criptosistema híbrido que combina técnicas de criptografía simétrica y criptografía asimétrica. Esta combinación permite aprovechar lo mejor de cada uno: El cifrado simétrico es más rápido que el asimétrico o de clave pública, mientras que éste, a su vez, proporciona una solución al problema de la distribución de claves en forma segura y garantiza el no repudio de los datos y la no suplantación. http://www.pgpi.org/doc/pgpintro/#p10.
[28] Dozecrypt, fue creada por la empresa australiana de software "Neuralabyss Software" para el cifrado de datos.
[29] Un certificado digital o certificado electrónico es un documento firmado electrónicamente por un prestador de servicios de certificación que vincula unos datos de verificación de firma a un firmante y confirma su identidad. (BOE, 2003)

2. CAPÍTULO SEGUNDO

"The ultimate security is your understanding of reality"

Howard Stanley Judd (New York,1936)

Cuando la gente oye hablar de Sistemas de Detección de Intrusos, generalmente los asocia a “alarmas de ladrones para ordenadores o redes”. Es fácil entender un concepto como este, usando comparaciones sencillas. En realidad, la explicación es bastante aproximada, y los usuarios que no se dedican a la seguridad no necesitan saber más. Sin embargo, los expertos en seguridad no pueden cometer el error de conformarse con algo trivial, sin tener conocimiento alguno sobre la historia de estos sistemas. En el presente capítulo serán descritos algunos fundamentos de los Sistemas de detección de intrusos.

2.1 CONCEPTOS DE DETECCIÓN DE INTRUSOS

A continuación se describirán los términos fundamentales de detección de intrusos y que serán utilizados frecuentemente a lo largo de la presente memoria:

Attack
*“RFC 2828, an assault on system security that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.”****[30]***

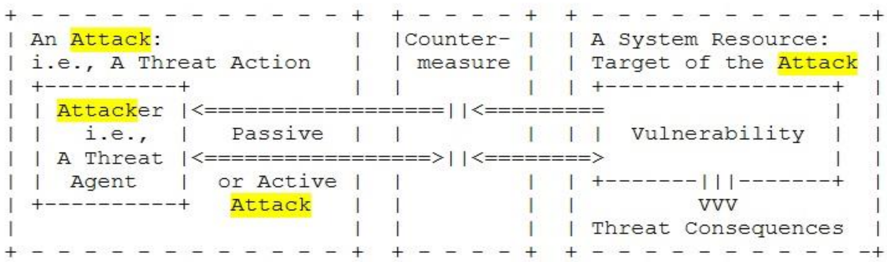


Figura 2.1 Se muestra el término ataque relacionado con otros conceptos básicos de seguridad [30].

Intruder

*“RFC 2828, an entity that gains or attempts to gain access to a system or system resource without having authorization to do so.”****[31]***

---

[30] RFC 2828, Ataque: es el asalto a la seguridad de un sistema que tiene su origen en una amenaza inteligente; esto es, un acto intelectual que debe considerarse como un deliberado intento (especialmente en la medida en que se utiliza un método o una técnica) de evadir los servicios de seguridad y violar la política de seguridad de dicho sistema.
[31] RFC 2828, Intruso: entidad que consigue o intenta conseguir acceso a un sistema o recurso de sistema sin tener autorización para hacerlo.

Security Intrusion

"RFC 2828, a security event, or a combination of multiple security events, that constitutes a security incident in which an intruder gains, or attempts to gain, access to a system (or system resource) without having authorization to do so."[32]

Intrusion Detection

"RFC 2828, a security service that monitors and analyzes system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner."[33]

Fortaleciendo los conceptos anteriores de la RFC 2828, existe una excelente defición en el libro "Intrusion Detection" de Rebecca Gurley Base :

"Detección de Intrusiones es el proceso de monitorizar los eventos que se dan en un sistema de computación o en una red de comunicaciones, y analizarlos en busca de signos de intrusión; definidos éstos como intentos de poner en compromiso la confidencialidad, integridad o disponibilidad, o de evadir los mecanismos de seguridad de dicho sistema o red. Dichas intrusiones son causadas por atacantes que consiguen acceder a sistemas desde Internet, por usuarios autorizados de dichos sistemas que intentan obtener privilegios adicionales para los cuales no están autorizados, o por usuarios autorizados que hacen un mal uso de los privilegios que se les han otorgado. Por otro lado, los Sistemas de Detección de Intrusiones (IDS) son productos software o hardware que automatizan este proceso de monitorización y análisis."

Con los conceptos previamente definidos, podemos hacernos la siguiente pregunta: ¿Por qué utilizar un IDS?, Sin la utilización de un IDS el administrador de una red no será capaz de identificar algunos ataques, pues muchas actividades maliciosas no dejan indicios de invasión, a no ser que el mismo administrador de la red dañe el sistema.

2.2 SISTEMA DE DETECCIÓN DE INTRUSOS (IDS)

Las intrusiones pueden ser producidas por los atacantes que tienen acceso al sistema, por usuarios autorizados que intentan obtener privilegios no autorizados, o por los usuarios autorizados que hacen el mal empleo de los privilegios brindados. La complejidad de tales situaciones aumenta en el caso de sistemas distribuidos de redes y en redes inseguras. Cuando los atacantes tratan de tener acceso a un sistema de red externo como el Internet, uno o varios hosts pueden estar implicados.

En algunos casos el atacante puede hacer uso de máquinas remotas que

[32] RFC 2828, Evento de seguridad, o combinación de múltiples eventos de seguridad, que constituye un incidente de seguridad en el cual un intruso consigue, o intenta conseguir, acceso a un sistema (o recurso de sistema) sin tener autorización para hacerlo.
[33] RFC 2828, Servicio de seguridad que monitoriza y analiza los eventos del sistema con el propósito de encontrar intentos de acceso a recursos de sistema de forma no autorizada, y proporcionar aviso en tiempo real (o cerca del tiempo real) sobre ellos.

fueron afectadas y que serán utilizados en ataques a otras redes.

Sin embargo, una intrusión puede implicar también a numerosos intrusos, que a varios objetivos de sistemas víctimas. Desde la perspectiva de la víctima, las intrusiones son caracterizadas por sus manifestaciones, que podrían o no incluir el daño. Algunos ataques no pueden producir ninguna manifestación mientras que algunas manifestaciones evidentes pueden ser producidas por el mal funcionamiento de la red o del sistema.

Por lo general, las responsabilidades de todo Sistema de Detección de Intrusiones son la captura de datos del sistema de información, el análisis de dichos datos en el intento de detectar algún tipo de actividad subversiva a partir de ellos, y el proporcionar una respuesta ante aquellas situaciones susceptibles de comprometer la seguridad del sistema. De esta forma, un primer modelo general de funcionamiento quedaría compuesto por estos tres pilares fundamentales: Recogida de información, Análisis y Respuesta. (LUNDIN, 2004; CERT/IN03; GONZÁLEZ A, 2003; ALESSANDRI, 2001; BACE, 2001)



Figura 2.2 Modelo general de funcionamiento del IDS (GARCÍA, 2006).

Las siguientes actividades son realizadas por los IDS en grados variables de precisión:

a) Monitorización y análisis de actividades de sistema y de los utilizadores.

b) Realización de auditorías a la infraestructura de la red, a las fallas y a las vulnerabilidades del sistema.

c) Identificación del modelo de actividades del sistema, en la manera de reconocer señales de ataques y de alertas.

d) Realización de un análisis estadístico a un modelo de comportamiento anómalo.

e) Validación de la integridad de los datos y de los sistemas.

f) Realización de auditorías de gestión a los sistemas de información, y de reconocimiento del comportamiento de los usuarios que desobedecen a la política de seguridad de la organización.

Existen IDS capaces de reaccionar ante la detección de acciones no autorizadas, conteniendo o deteniendo el ataque. Por ejemplo cortando la conexión a la red. Las técnicas de detección de intrusos presumen que el comportamiento y las actividades de un usuario o programa legítimo son diferentes del comportamiento y de las actividades de un invasor.

Los IDS funcionan con base en el análisis y clasificación de los comportamientos y actividades de un usuario como legítimos o ilegítimos. Es decir; esas técnicas también asumen que el comportamiento y las actividades de un usuario son observables y posibles de ser analizadas.

2.2.1 TAXONOMÍA DE LOS IDS

Los IDS pueden ser clasificados sobre la base de varios rasgos diferentes (McHUGH, KEMMERER, DASGUPTA), aunque sólo los más característicos serán usados en esta sección. Los subconjuntos de estos rasgos, que se refieren a los funcionamientos internos del IDS pueden ser definidos como: método de detección (detection method), supervisando el alcance (monitoring scope), comportamiento sobre la detección (behavior on detection) y tiempo de ejecución. Una clasificación estándar de los IDS, basado en su método de detección, o el modelo de intrusiones, define los paradigmas siguientes:

2.2.1.1 DETECCIÓN DE USOS INDEBIDOS (MISUSE-BASED ID):

También conocido como detección basada en conocimiento (Knowledge-based ID), “Detection method that analyzes system activity, looking for events or sets of events that match a predefined pattern of events that describe a known attack…”**[34]**

Se caracteriza por su elevada precisión a la hora de decidir la legitimidad de una determinada acción. Sin embargo, adolece de ciertos problemas, entre los cuales el más relevante es su absoluta falta de respuesta ante ataques aún no documentados. Un intruso puede construir una ligera variante de un ataque clásico, o por supuesto un ataque completamente nuevo, y evadir así la vigilancia del Sistema de Detección de Intrusiones. (GARCÍA, 2006).

- **VENTAJAS**

  - Precisión en la detección: Los Sistemas de Detección de Usos Indebidos son muy efectivos en la detección de los ataques que tienen documentados. Esto es debido a que su principio de funcionamiento se reduce a la comparación de la actividad cotidiana del sistema que se pretende proteger, con una base de firmas de ataques que se construye a partir de vasto conocimiento experto sobre métodos de intrusión y procedimientos para explotar vulnerabilidades conocidas.

---

[34] Método de detección que analiza la actividad del sistema, en busca de eventos o conjuntos de eventos que concuerden con un patrón de eventos predefinido que describe un ataque conocido.

De esta forma, cualquier actividad subversiva que se encuentre tipificada en dicha base de conocimiento es advertida rápidamente. (LUNDIN, 2004; GOLOMB, 2003; Holz03; BACE, 2001; AXELSSON, 2000; DEBAR, 1999)

- Ausencia de falsos positivos: Dada la simplicidad del método de detección, no es habitual que se produzca el fenómeno del falso positivo, siempre y cuando la base de conocimiento se encuentre convenientemente configurada. Ésta es precisamente la mayor ventaja que presenta este tipo de sistemas de detección con respecto a otros, como puede ser la Detección de Anomalías, que ha sufrido tradicionalmente este problema (LUNDIN, 2004; GONZÁLEZ A, 2003; BACE, 2001; ALLEN, 2000; DEBAR, 1999; LINDQVIST, 1999).

- Estabilidad del conocimientos: Una característica negativa que presentan algunos Sistemas de Detección de Intrusiones más ambiciosos que los basados en la Detección de Usos Indebidos, es la posibilidad de que los modelos de representación de conocimiento adaptativos que utilizan puedan ser dirigidos progresiva e intencionadamente para que, llegado el momento, dejen pasar desapercibidos ciertos eventos que el potencial intruso no desea que sean detectados. Si el sistema de detección es capaz de adaptarse a los cambios de comportamiento de los usuarios, es posible que alguno de esos usuarios utilice esa indeseable característica con fines maliciosos. Sin embargo, en el caso de la detección de firmas, la sencilla estructura de su motor de decisión elimina este problema. (LUNDIN, 2004; GONZÁLEZ A, 2003; ALESSANDRI, 2001; BACE, 2001; DEBAR, 1999; JACKSON, 1999)

- Bajo coste administrativo: Un Sistema de Detección de Usos Indebidos puede proporcionar de forma rápida y fiable un diagnóstico exacto sobre la situación de un sistema de información, sin necesidad de laboriosos procedimientos administrativos y sin necesidad de disponer de personal de seguridad excepcionalmente experimentado (LUNDIN, 2004; GOLOMB, 2003; BACE, 2001; ALLEN, 2000; DEBAR, 1999; LINDQVIST, 1999).

- **DESVENTAJAS**

  - Incapacidad de detección de ataques no documentados: La limitación por excelencia de todo sistema de detección de usos indebidos, dada la naturaleza del propio principio de funcionamiento, consiste en la incapacidad absoluta de este tipo de sistemas de advertir cualquier comportamiento subversivo que no se encuentre documentado en su base de conocimiento. Dado el extraordinario número de vulnerabilidades que se encuentran diariamente, así como de procedimientos de ataque que las explotan [CERT/CC05], esta cuestión adquiere de forma natural carácter de urgencia. Actualmente, las soluciones al problema pasan por la actualización automática periódica de las bases de firmas, si bien este método termina por reducir la eficiencia del sistema de detección a medio y largo plazo. (LUNDIN, 2004; GONZÁLEZ A, 2003; Julisch03; ALESSANDRI, 2001; BA-

CE, 2001; ALLEN, 2000; AXELSSON, 2000; DEBAR, 1999; JACKSON, 1999, LINDQVIST, 1999).

- Rendimiento decreciente: El crecimiento que se produce en las bases de conocimiento a medida que se detectan, analizan y documentan nuevos tipos de ataques, hace que dichas bases sean complejas de mantener, y que el rendimiento del sistema termine por degradarse. En esta situación, el sistema de detección puede llegar incluso a saturarse, y a tomar la decisión de ignorar aquellos eventos que están saturando sus mecanismos de control de la congestión. Por supuesto, si el sistema se ve obligado a dejar de analizar eventos, su eficacia pasa a verse en un serio compromiso. De hecho, existen técnicas de evasión de Sistemas de Detección de Intrusiones [Ptacek+98] que se basan en esta indeseada característica (LUNDIN, 2004; GOLOMB, 2003; DEBAR, 1999; JACKSON, 1999).

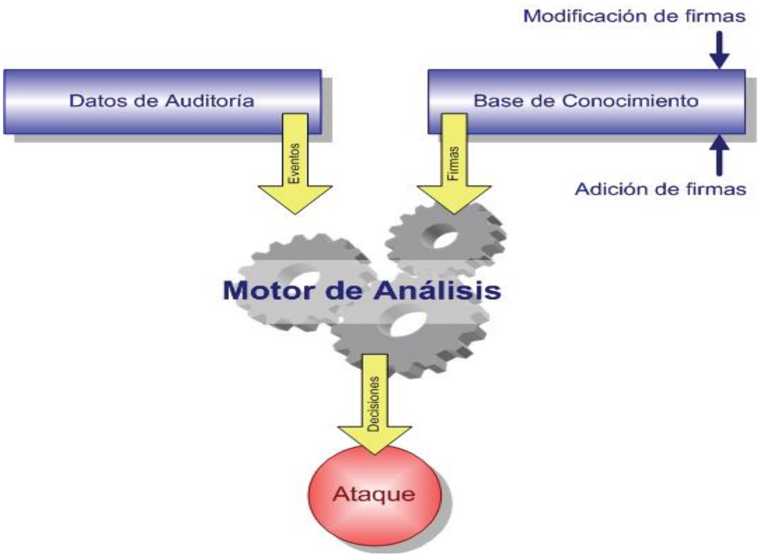


Figura 2.3 Modelo general de Detección de Usos Indebidos (GARCÍA, 2006).

2.2.1.2 DETECCIÓN DE ANOMALÍAS (ANOMALY-BASED ID)

También conocido como detección basada en comportamiento (*behaviour-based ID*), *"Detection method that identifies abnormal unusual behavior (anomalies) on a host or network…"* ***[35]***

Con el objetivo de aportar una solución al principal problema de los sistemas de detección de uso indebidos, y proporcionar un método capaz de detectar ataques desconocidos hasta el momento surge la detección de anomalías (GARCÍA, 2006), esta filosofía de detección también está basada en técnicas de Inteligencia Artificial, pero plantea una forma de análisis complementaria a la anterior. En lugar de utilizar el conocimiento subversivo conocido para llevar a cabo el proceso de decisión, utiliza el conocimiento legítimo **[36]**. De esta forma, a partir de las acciones habituales de Los usuarios, se construye un perfil de comportamiento que posteriormente es

[35] Método de detección que identifica comportamiento inusual o anormal (anómalo) en una máquina o en una red.
[36] En este caso se está usando política de lista blanca o whitelist.

contrastado con la actividad cotidiana que se registra durante la explotación normal del sistema.

Así, es posible detectar acciones que se salen de la forma de proceder habitual de dichos usuarios; acciones entre las cuales se encuentran los novedosos modos de ataque mencionados anteriormente. A este respecto, es importante destacar el hecho de que con este modo de operación se consigue que el proceso de detección sea completo (DEBAR, 1999); característica de la que adolecen los sistemas de Detección de Usos Indebidos. Por lo general, este modelo de funcionamiento está basado en las propiedades estadísticas de los diferentes eventos que se producen en el sistema. De esta forma, es posible que el sistema de detección *aprenda* la naturaleza del comportamiento de los usuarios del sistema, e infiera sus conclusiones, por lo general construidas en base al concepto de probabilidad. Este modo de operación hace que el sistema pueda prescindir de todo conocimiento apriorístico, y que pueda detectar ataques para los cuales no existe un modelo previamente documentado. Sin embargo, su naturaleza probabilística y la necesidad del sistema de detección de permanecer en un estado de adaptación continua a las variaciones del comportamiento de los usuarios, hace a dichos sistemas susceptibles de generar falsos positivos. Esta característica no es preocupante siempre que se mantenga dentro de unos límites de excepcionalidad, pero puede convertirse en un serio problema si se da con mayor frecuencia. De hecho, la minimización de los falsos positivos es uno de los mayores retos que actualmente presenta esta tecnología. (GARCÍA, 2006 ; LUNDIN, 2004; GONZÁLEZ A, 2003; BACE, 2001; ALLEN, 2000; AXELSSON, 2000; DEBAR, 1999; JACKSON, 1999; LINDQVIST, 1999). Este enfoque proporciona capacidad de respuesta ante los peligrosos Zero Day Attacks y llega más allá de las habilidades propias de los Sistemas de Detección de Usos Indebidos. No obstante, también presenta un importante inconveniente: Un nuevo elemento o componente en el sistema introduce de forma natural nueva actividad, la cual no deja de ser legítima y, sin embargo, es calificada por el sistema de detección como subversiva (ALLEN, 2000; AXELSSON, 2000). Esta cuestión obliga a una actualización continua del perfil de comportamiento, y hace que el sistema presente una indeseada característica conocida como falso positivo **[37]**. (GARCÍA, 2006)

- **VENTAJAS**

  - Detección de ataques desconocidos *(Zero day attacks*)
    La mayor capacidad de este tipo de Sistemas de Detección de Intrusiones es precisamente la de poder responder ante lo desconocido. Dado que su filosofía de funcionamiento está basada en el aprendizaje que el propio sistema de detección hace sobre la realidad de la infraestructura hardware y software que pretende proteger, es este conocimiento aprendido el elemento que marca la diferencia entre lo legítimo y lo subversivo. De esta forma, no es necesario documentar todos y cada uno de los tipos de ataque que aparecen diariamente, sino que basta con realizar un ejercicio de introspección que determine cómo se comporta habitualmente un sistema de

[37] Evento incorrectamente identificado como intrusión por el Sistema de Detección de Intrusiones, cuando por el contrario no ha ocurrido tal intrusión.

información, y cómo no (LUNDIN, 2004; GOLOMB, 2003; GONZÁLEZ A, 2003; BACE, 2001; ALLEN, 2000; AXELSSON, 2000; DEBAR, 1999; JACKSON, 1999; LINDQVIST, 1999).

- Documentación de ataques desconocidos
  Dada la poderosa propiedad anterior, un sistema de detección basado en anomalías puede generar automáticamente información descriptiva sobre los nuevos ataques que detecta. De esta forma, es posible construir, en base a dicha información, nuevas firmas de ataques que pueden ser utilizadas por los sistemas de detección de usos indebidos, de cara a la utilización conjunta de ambos. (BACE, 2001)

- Rendimiento constante
  El modelo de representación de conocimiento de este tipo de sistemas no necesita incrementar su volumen de conocimiento, sino simplemente adaptarlo a ajustarlo a la realidad de la infraestructura subyacente. Esta propiedad hace que estos sistemas tengan un prometedor futuro, ya que no adolecen del problema de la pérdida de eficiencia que sufren los modelos de detección de usos indebidos (LUNDIN, 2004; GOLOMB, 2003).

- **DESVENTAJAS**

  - Inestabilidad del conocimiento
    Dada la necesidad de este tipo de sistemas de detección de adaptación continua a la siempre cambiante realidad del sistema a proteger, el conocimiento adquirido por los Sistemas de Detección de Anomalías presenta cierta propensión a la inestabilidad. El comportamiento del sistema cambia, y ese cambio puede influir para que lo que en un momento dado ha podido ser considerado como legítimo, ahora pase a ser subversivo, y viceversa: que lo que en el pasado era considerado anómalo, pase a convertirse en habitual. Por supuesto, esta cuestión se da cerca de los umbrales de detección, y no significa que el sistema sufra de indeterminismo (LUNDIN, 2004; BACE, 2001; AXELSSON, 2000; DEBAR, 1999; JACKSON, 1999; LINDQVIST, 1999).

  - Posibilidad de subversión del conocimiento
    Debido a la misma necesidad de adaptación del apartado anterior, aparece una remota, aunque posible, probabilidad de que el inherente proceso de aprendizaje de este tipo de sistemas sea utilizado subversivamente, con el objetivo de que en un momento del futuro el sistema de detección deje de percibir como anómalos ciertos eventos maliciosos.

    Para ello, un intruso puede provocar a lo largo de un cierto periodo de tiempo ciertos sucesos normales (legítimos), pero cuyos atributos cada vez se van acercando progresivamente al terreno de lo anómalo. De esta forma, teóricamente, llegaría un momento en el futuro en el que el sistema, que se habría adaptado a esos cambios, consideraría un evento antes sub-

versivo como perfectamente legítimo. (LUNDIN, 2004; GONZÁLEZ A, 2003; ALESSANDRI, 2001; BACE, 2001; DEBAR, 1999; JACKSON, 1999)

- Alto coste administrativo
  En la actualidad, el proceso de aprendizaje que caracteriza a los sistemas de detección de anomalías aún no ha conseguido liberarse de una relativamente costosa tarea de configuración y administración. Asimismo, y dada la naturaleza cambiante de dichos sistemas, el mantenimiento de los mismos también contribuye de forma notable a incrementar los esfuerzos que una organización necesita soportar para que su sistema de detección pueda ser explotado con las mayores garantías. Además, dicho proceso de aprendizaje no sólo precisa de esas importantes tareas de configuración y administración, sino que, además, los modelos computacionales que se utilizan (como pueden ser las redes neuronales o los algoritmos genéticos) requieren muy frecuentemente la supervisión del mismo por parte del operador humano. (LUNDIN, 2004; BACE, 2001; ALLEN, 2000; JACKSON, 1999)

- Presencia de falsos positivos
  Debido fundamentalmente a la naturaleza probabilística de los modelos de Detección de Anomalías, es frecuente la necesidad de tomar un compromiso sobre la magnitud del volumen de respuestas que se consideran convenientes para su correcto procesamiento posterior. Dicho compromiso se establece habitualmente en base a valores umbral que separan lo que se considera normal (y por lo tanto, legítimo) de lo que se considera anómalo (y por lo tanto, ilegítimo).

  Un umbral muy restrictivo hace que cualquier desviación, por leve que sea, sea considerada merecedora del lanzamiento de una alarma u otra acción de respuesta. En principio, la seguridad del sistema es muy alta, y se responde ante cualquier evento sospechoso. Sin embargo, el volumen de respuestas es enorme, y en el caso habitual de utilizarse la simple respuesta pasiva mediante notificaciones de alarma, el administrador queda automáticamente saturado de ingentes cantidades de mensajes que no puede procesar y que por lo general sólo implican acciones legítimas pero ligeramente desviadas del perfil de comportamiento. O lo que es lo mismo: falsos positivos. Además, y lo que es aún peor, entre esos grandes volúmenes de mensajes suelen pasar desapercibidos auténticos peligros, los cuales no pueden ser físicamente diferenciados de los anteriores. Por el contrario, un umbral muy permisivo genera un número razonable de respuestas, pero posibilita la no detección de ataques cercanos a dicho umbral, dando origen a los aún más peligrosos falsos negativos.

  Por todo ello, parece evidente la necesidad de buscar un equilibrio entre los dos extremos, que permita minimizar ambos efectos indeseados, si bien su eliminación completa es improbable (LUNDIN, 2004; GOLOMB, 2003; GONZÁLEZ A, 2003; BACE, 2001; Allen, 2000; AXELSSON, 2000; DEBAR, 1999; JACKSON, 1999; LINDQVIST, 1999).

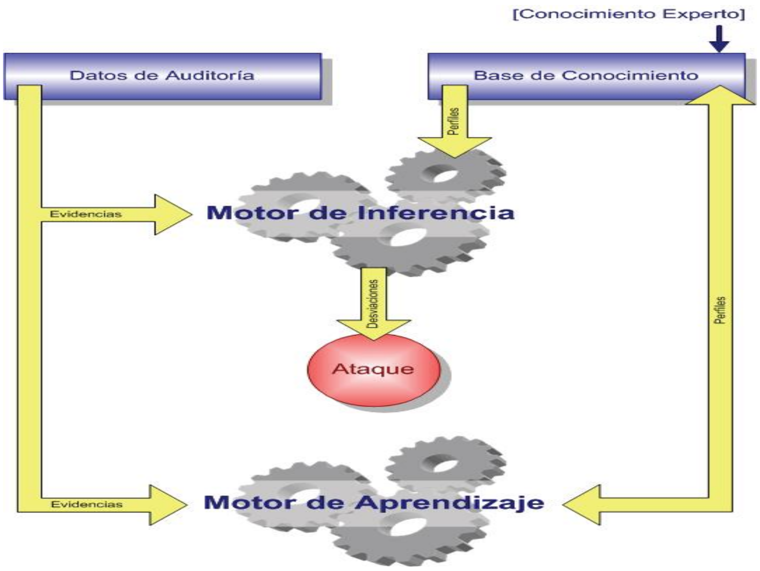


Figura 2.4 Modelo general de Detección de Anomalías (GARCÍA, 2006).

2.2.1.3 DETECCIÓN EN ESPECIFICACIONES (SPECIFICATION-BASE ID)

Este tercer paradigma fue presentado en (BALEPIN, 2003). Esto confía en comportamientos específicos reflejados en políticas de sistemas que son usados como una base para descubrir ataques. Esta técnica ha sido propuesta como una alternativa prometedora que combina la fuerza de los dos anteriores.

La Clasificación de los IDS en cuanto a la fuente de información, supervisando el alcance (monitoring scope), también conocido como audit source location o data source define lo siguiente (CASWEL, 2003; FRANCESQUINI, 2004):

2.2.1.4 HOST-BASED IDS (HOST INTRUSION DETECTION SYSTEM -HIDS)

Son sistemas que monitorean a un determinado computador (Host), buscando detectar anomalías que indican un riesgo potencial, revisando las actividades en el computador (Host). Normalmente, estos sistemas monitorizan variables como la tasa de utilización del procesador, llamadas a funciones de sistemas y modificación en archivos críticos, en búsqueda de actividades sospechosas. Las funciones de este tipo de software son muy similares a las de los IDS. Configuraciones típicas permiten varios HIDS repartidos por la red que envían sus resultados a un servidor centralizado que los analizará en busca de los riegos y alertas antes mencionados.

Este fue el primer tipo de software de detección de intrusos que se diseñó, siendo el objetivo original el Mainframe[38], donde la interacción exterior era infrecuente.**[39]**

[38] Una computadora central (en inglés *mainframe*) es una computadora grande, potente y costosa usada principalmente por una gran compañía para el procesamiento de una gran cantidad de datos; por ejemplo, para el procesamiento de transacciones bancarias.
[39] «Towards a taxonomy of intrusion-detection systems». *Computer Networks* **31** (8): pp. 805–822. April, 23 1999. http://www.sciencedirect.com/science/article/pii/S1389128698000176.

Los HIDS generalmente hacen todo lo posible para evitar que las bases de datos de objetos, de checksum y sus reportes sufran cualquier forma de manipulación. Después de todo, si los intrusos lograran modificar cualquiera de los objetos que los HIDS monitorean, nada podría detener a los intrusos de la modificación de este tipo a los propios HIDS - a menos que los administradores de seguridad tomen las precauciones adecuadas. Muchos gusanos[40] y virus[41] a tratar de desactivar las herramientas anti-virus, por ejemplo.

Aparte de las cripto-técnicas, los HIDS podrían permitir a los administradores almacenar las bases de datos en un CD-ROM o en otras dispositivos de memoria de sólo lectura (otro factor de lucha para las actualizaciones poco frecuentes ...) o almacenarlos en una memoria fuera del sistema. Del mismo modo, un HIDS frecuentemente envía sus registros (logs) fuera del sistema de inmediato - por lo general utilizando canales VPN a algún sistema de gestión central.

Se podría argumentar que el módulo de plataforma de confianza es un tipo de HIDS. A pesar de su alcance difiere en muchos aspectos a la de un HIDS, fundamentalmente, proporciona un medio para identificar si hay algo/alguien que ha manipulado una porción de un ordenador. Arquitectónicamente este proporciona la máxima (por lo menos hasta ahora) detección de intrusiones basado en host, ya que depende de un hardware externo a la CPU en sí, por lo tanto por lo que es mucho más difícil para un intruso corromper a sus bases de datos de objetos y de checksums. InfoWorld[42] opina que el software HIDS es una forma útil para administradores de red de encontrar malware, sugiriendo que debería usarse en todos los servidores, no solo en los servidores críticos.

- **VENTAJAS**

Las principales ventajas del HIDS son:

- Herramienta potente, registra comandos utilizados, ficheros abiertos,...
- Tiende a tener menor número de falsos-positivos que los NIDS, entendiendo falsos-positivos a los paquetes etiquetados como posibles ataques cuando no lo son.
- Menor riesgo en las respuestas activas que los IDS de red.

- **DESVENTAJAS**

Como principales desventajas se encuentran:

- Requiere instalación en la máquina local que se quiere proteger, lo que supone una carga adicional para el sistema.
- Tienden a confiar en las capacidades de auditoria y logging de la máquina en sí.

[40] Un gusano informático (también llamado IWorm por su apócope en inglés, I de Internet, Worm de gusano) es un malware que tiene la propiedad de duplicarse a sí mismo.
[41] Un virus informático es un malware que tiene por objeto alterar el normal funcionamiento de la computadora, sin el permiso o el conocimiento del usuario.
[42] InfoWorld es un medio de comunicación en línea y negocio de eventos dedicado a Tecnologías de Información que forma parte de InfoWorld Media Group, una división de IDG (International Data Group).

2.2.1.5 NETWORK-BASED IDS (NETWORK INTRUSION DETECTION SYSTEM NIDS)

Son sistemas que monitorizan segmentos de redes de computadores. Normalmente, estos sistemas capturan todos los paquetes de datos de tráfico en determinados segmentos de la red en la búsqueda de actividades sospechosas. Busca detectar anomalías que inicien un riesgo potencial, tales como ataques de denegación de servicio, escaners de puertos o intentos de entrar en un ordenador, analizando el tráfico en la red en tiempo real. Para ello, analiza todos los paquetes, buscando en ellos patrones sospechosos. Los NIDS no sólo vigilan el tráfico entrante, sino también el saliente o el tráfico local, ya que algunos ataques podrían ser iniciados desde el propio sistema protegido. A pesar de la vigilancia, su influencia en el tráfico es casi nula.

Para que los NIDS sean efectivos, han de ser actualizados periódicamente. En caso de detectar un ataque contra el sistema, puede tomar medidas protectoras. Un aspecto negativo de los NIDS actuales es su complicación a la hora de obtener las opciones de configuración óptimas para su ejecución. De otro modo, obtendremos demasiados falsos positivos (falsas alarmas, con gran cantidad de información que luego un administrador tendrá que procesar) o pasará sin advertir ciertos ataques. Los NIDS tienen dos componentes:

- Un sensor: situado en un segmento de la red, la monitoriza en busca de tráfico sospechoso
- Una Consola: recibe las alarmas del sensor o sensores y dependiendo de la configuración reacciona a las alarmas recibidas.

- **VENTAJAS**

Las principales ventajas del NIDS son:

- Detectan accesos no deseados a la red.
- No necesitan instalar software adicional en los servidores en producción.
- Fácil instalación y actualización por que se ejecutan en un sistema dedicado.

- **DESVENTAJAS**

Como principales desventajas se encuentran:

- Examinan el tráfico de la red en el segmento en el cual se conecta, pero no puede detectar un ataque en diferentes segmentos de la red. La solución más sencilla es colocar diversos sensores.
- Pueden generar tráfico en la red.

- Ataques con sesiones encriptadas son difíciles de detectar.

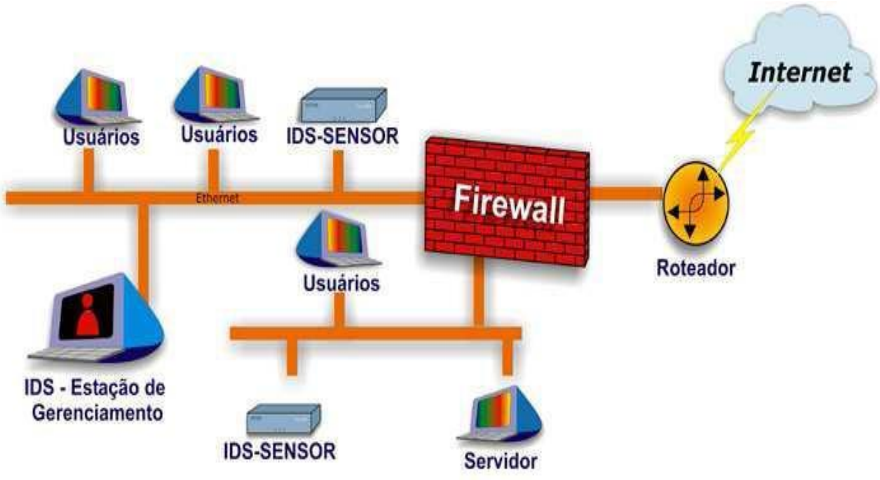


Figura 2.5 IDS basado en Red (NIDS)

2.2.1.6 MULTIHOST-BASED IDS

Analiza datos de múltiples hosts es usado para detectar intrusiones. Los hosts de un segmento de red pueden ser monitoreados por estos IDS.

2.2.1.7 HOST/NETWORK-BASED IDS

Conocido también como sistema de detección de intrusión distribuido-DIDS, son sistemas que funcionan con sensores remotos, su diseño viene a ser la combinación de HIDS y de NIDS. Este tipo de IDS propone el aprovechar al máximo las ventajas de estos dos tipos de IDS. Uno de los problemas claves de esta fusión es el desarrollo de una interfaz que pueda mostrar toda la información distribuida capturada por los sensores en la red.

La Clasificación de los IDS basado en comportamientos (behaviour):

2.2.1.8 PASIVO

Los primeros IDS eran pasivos, se limitaban a informar de los intentos de intrusión al administrador. De poco sirve detectar un ataque para que horas después el administrador reciba un mensaje que informe de que se vio la intrusión pero no se intentó hacer nada por abortarla.

2.2.1.9 ACTIVO

Son capaces de tomar acciones correctivas orientadas a detener ataques en el mismo instante en que se producen.

La clasificación de los IDS basado según el tiempo de ejecución, pueden ser periódicos o de tiempo real. Así como los NIDS suelen dar respuesta en tiempo real, los primeros HIDS se ejecutaban periódicamente para buscar indicios de intrusión. Después se fue reduciendo el intervalo entre la ocurrencia del evento y su análisis, hasta el punto que es posible gestionar los eventos en el instante de su registro. Los sistemas de red implementados como parte de la pila de red de las maquinas protegidas ofrecen las mismas prestaciones de respuesta inmediata (con posibilidad de cancelación de conexiones) que los NIDS

2.2.2 LIMITACIONES DE LOS IDS

En los sistemas de detección de intrusión pueden ocurrir posibles errores que son clasificados como falsos positivos, falsos negativos o error de subversión.

2.2.2.1 FALSOS POSITIVOS

Evento incorrectamente identificado como intrusión por el Sistema de Detección de Intrusiones, cuando por el contrario no ha ocurrido tal intrusión. Como formas de reducir la ocurrencia de falsos positivos, encontramos: (STEFFEN, 2003)

- Ajustar los niveles de prioridad de las reglas, conforme a la estructura y características de la red.
- Desactivar reglas referentes a servicios que no son ejecutados en la red.
- Mejorar o crear nuevas reglas con base en la estructura y en las características de la red.

2.2.2.2 FALSOS NEGATIVOS

Evento no identificado como intrusión por el Sistema de Detección de Intrusiones, cuando por el contrario sí es merecedor de dicha identificación. La ocurrencia de falsos negativos es extremadamente más critica que la ocurrencia de falsos positivos, pues de hecho que el sistema está sobre ataque y la herramienta no la detecta. Generalmente los falsos negativos ocurren por causa: de la sobrecarga de la red, de un ataque nuevo o desconocido, de una configuración herrada en el IDS (STEFFEN, 2003).

Además de la existencia de falsos positivos y negativos en los IDS, es común la ocurrencia de errores de subversión. Error de subversión ocurre cuando el IDS es modificado por una intrusión a fin de forzar la ocurrencia de falsos negativos (MARTIN, 2006)

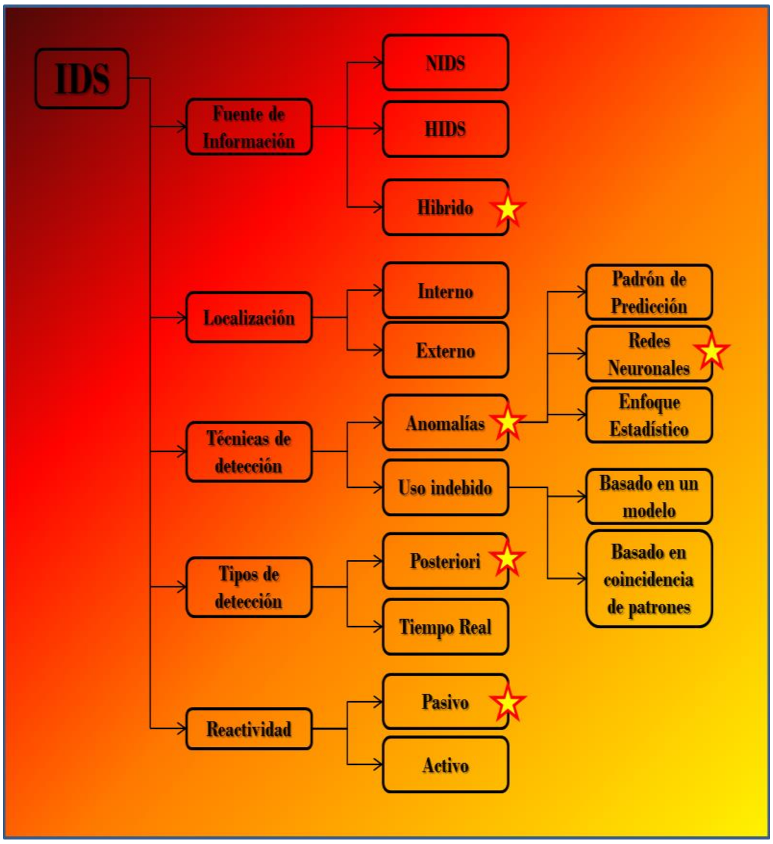


Figura 2.6 Clasificación de los IDS

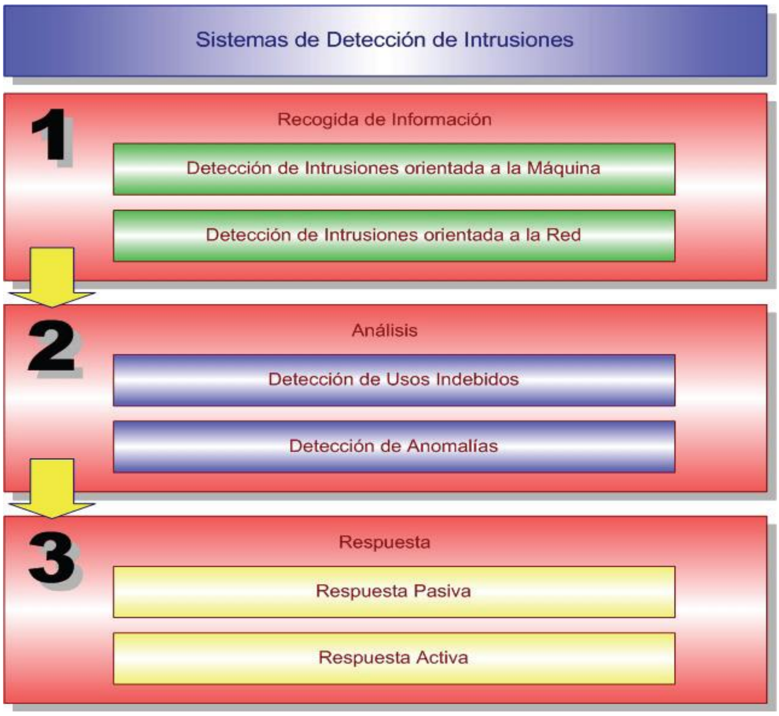


Figura 2.7 Taxonomía general de los IDS (GARCIA, 2006).

2.2.3 HONEYPOTS

El término *honeypot* significa, literalmente, 'tarro de miel'. Se denomina *honeypot* al software o conjunto de computadores cuya intención es atraer a atacantes, simulando ser sistemas vulnerables o débiles a los ataques. Es una herramienta de seguridad informática utilizada para recoger información sobre los atacantes y sus técnicas. Los *honeypots* pueden distraer a los atacantes de las máquinas más importantes del sistema, y advertir rápidamente al administrador del sistema de un ataque, además de permitir un examen en profundidad del atacante, durante y después del ataque al *honeypot*.

> *"Um honeypot é um sistema que possui falha de seguranças reais ou virtuais, configuradas de maneira proposital, a fim de ser invadido para que no futuro os resultados desses ataques possam ser estudados. Um honeypot é uma ferramenta de estudo de segurança da informação, em que seu objetivo é colher informações do atacante. Os honeypots não fazem nenhum tipo de prevenção, mas oferecem informações adicionais de valor inestimável."* **[43]** *(WEISSHEIMER, 2006).*

Algunos honeypots son programas que se limitan a simular sistemas operativos no existentes en la realidad y se les conoce como *honeypots* de baja interacción y son usados fundamentalmente como medida de seguridad. Otros sin embargo trabajan sobre sistemas operativos reales y son capaces de reunir mucha más información; sus fines suelen ser de investigación y se los conoce como *honeypots* de alta interacción.

Un tipo especial de honeypot de baja interacción son los *sticky honeypots* (*honeypots* pegajosos) cuya misión fundamental es la de reducir la velocidad de los ataques automatizados y los rastreos. En el grupo de los honeypot de alta interacción nos encontramos también con los *honeynet*.

En 2006, la BBC realizó una investigación para determinar la incidencia de ataques que podría sufrir un ordenador típico. En una primera fase, consistente en simplemente registrar el número y tipo de ataques, los resultados fueron que cada 15 minutos, como media, el ordenador recibía spam «molesta», típicamente ofertas fraudulentas para mejorar la seguridad del ordenador. Sin embargo, cada hora, como media, el ordenador recibía ataques más serios, de gusanos informáticos tipo SQL.Slammer y MS.Blaster.**[44]**

La segunda fase de la investigación consistía en dejar entrar el *adware* y *spyware* para ver sus eventuales efectos en el ordenador. El resultado fue que, de tratarse de un ordenador cualquiera en un hogar normal, hubiera quedado totalmente insegura e inservible.**[45]**

---

[43] "Un honeypot es un sistema que posee fallas de seguridad reales o virtuales, configuradas de manera intencional, a fin de ser invadido para que en el futuro los resultados de esos ataques puedan ser estudiados. Un honeypot es una herramienta de estudio de seguridad de la información, en que su objetivo es coger información del atacante. Los honeypots no hacen ningún tipo de prevención, pero ofrecen informaciones adicionales de valor inestimable."
[44] «Tracking down hi-tech crime» BBC News. http://news.bbc.co.uk/2/hi/technology/5414502.stm
[45] «Trapping hackers in the honeypot» BBC News. http://news.bbc.co.uk/2/hi/technology/6035455.stm

2.3 VISUALIZACION PARA LA SEGURIDAD EN LA RED

Las técnicas de visualización han sido aplicadas a masivos datasets por muchos años, son considerados un viable acercamiento a la busca de información, el ser humano es capaz de reconocer diferentes rasgos y detectar anomalías por medio de la inspección de gráficos (AHLBERG, 1999). La inspección visual del modelo de tráfico de la red es presentada como una alternativa crucial para la administración y monitoreo de la red (BECKER, 1995), como su objetivo principal, es proveer al administrador de la red de una representación sintética de la situación de su red.

Para realizar esta tarea, los instrumentos de visualización pueden:

- Ayudar a los administradores de la red en detectar anomalías y amenazas potenciales con una demostración intuitiva de la progresión del tráfico de la red.
- Tratar fácilmente con datos sumamente heterogéneos y ruidosos, tal como los datos que son requeridos para el monitoreo de la red y su identificación (CHOI, 2009).
- Proveer a los administradores de la red el soporte automatizado y motiva su efectividad explotando la capacidad del ojo humano de extrapolar el modelo del tráfico normal y detectar anomalías. Indicado en (CONTI, 2007), "un cuadro vale como miles de paquetes" o "un cuadro vale como miles logs de entrada".(MARTY, 2008)
- Ayuda a los administradores de red diagnosticar el funcionamiento de la red o al entendimiento del modelo de las comunicaciones entre nodos.
- Sirve como herramientas complementarias a otros mecanismos de seguridad.

La tarea de monitoreo que detecta acontecimientos intrusivos o anómalos puede ser alcanzado por la visualización de datos en los diferentes niveles de abstracción: nodos de red, alarmas de intrusión, niveles de paquetes de datos, etcétera. En otras palabras, las estadísticas de diferentes herramientas de seguridad pueden ser visualizadas, lo cual no sólo se enfocan en datos de tráfico, sino también sobre la topología de la red.

Hasta el momento, la mayor parte de IDSs han centrado sus estudios desde un solo punto de vista. Los IDSs realizan una clasificación de 2 clases de tráfico de red: normal/anómalo e Identifican en base a la anomalía (anomaly), en intrusivo/no intrusive y en base al mal uso (misuse). La presente memoria propone técnicas de visualización para la supervisión de red y su Identificación. Vale la pena acentuar

que la propuesta implica la visualización de datos de tráfico de red para descubrir intrusiones (es decir la visualización para identificar) y no la visualización de alarmas de IDS o logs (que es la visualización de identificar) como otros estudios lo han hecho. (ITOH, 2006)

En el cuarto capítulo de la presente memoria demostraremos la proyección satisfactoria de los modelos neuronales en la visualización del tráfico de red por la reducción de dimensionalidad.

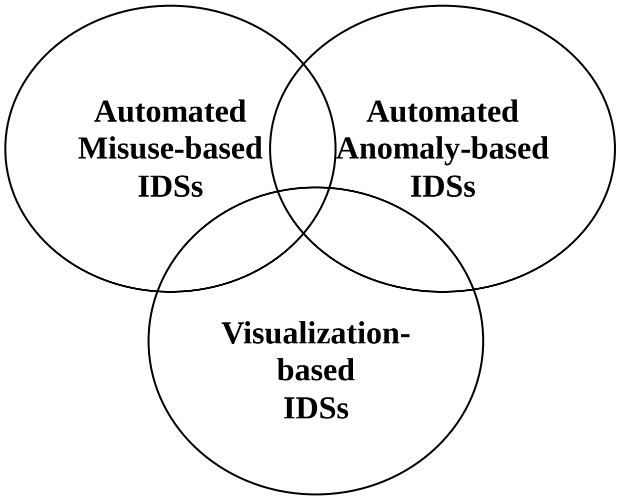


Figura 2.8 Mutually supporting capabilities of IDS

2.4 TÉCNICAS DE VISUALIZACIÓN

Las técnicas de visualización han sido ampliamente cubiertas en diversos artículos científicos; dos de los trabajos más relevantes que existen son (WILKINSON, 2005; KEIM, 2002). En esta sección detallaré brevemente algunas de las técnicas más comunes, que serán materia de estudio y aplicación en la presente memoria:

- Scatter plots (Diagrama de dispersión): Es un tipo de diagrama matemático que utiliza las coordenadas cartesianas para mostrar los valores de dos variables para un conjunto de datos.

  Los datos se muestran como un conjunto de puntos, cada uno con el valor de una variable que determina la posición en el eje horizontal y el valor de la otra variable determinado por la posición en el eje vertical. (UTSS, 2005) Un diagrama de dispersión se llama también gráfico de dispersión.

  Se emplea cuando una variable está bajo el control del experimentador. Si existe un parámetro que se incrementa o disminuye de forma sistemática por el experimentador, se le denomina parámetro de control o variable independiente = eje de x; y habitualmente se representa a lo largo del eje ho-

rizontal. La variable medida o dependiente = eje de y; usualmente se representa a lo largo del eje vertical. Si no existe una variable dependiente, cualquier variable se puede representar en cada eje y el diagrama de dispersión mostrará el grado de correlación[46] (no causalidad) entre las dos variables.

Un diagrama de dispersión puede sugerir varios tipos de correlaciones entre las variables con un intervalo de confianza determinado. La correlación puede ser positiva (aumento), negativa (descenso), o nula (las variables no están correlacionadas). Se puede dibujar una línea de ajuste (llamada también "línea de tendencia") con el fin de estudiar la correlación entre las variables. Una ecuación para la correlación entre las variables puede ser determinada por procedimientos de ajuste. Para una correlación lineal, el procedimiento de ajuste es conocido como regresión lineal y garantiza una solución correcta en un tiempo finito.

Uno de los aspectos más poderosos de un gráfico de dispersión, sin embargo, es su capacidad para mostrar las relaciones no lineales entre las variables. Además, si los datos son representados por un modelo de mezcla de relaciones simples, estas relaciones son visualmente evidentes como patrones superpuestos.

En la Figura 2.7 se muestra la utilización de la técnica del diagrama de dispersión, representando las dos primeras dimensiones (sepal longitud y sepal anchura) del bien conocido conjunto de datos Iris (IRIS DATASET, 1996). Los colores son también usados para distinguir las tres diferentes clases de conjuntos de datos.

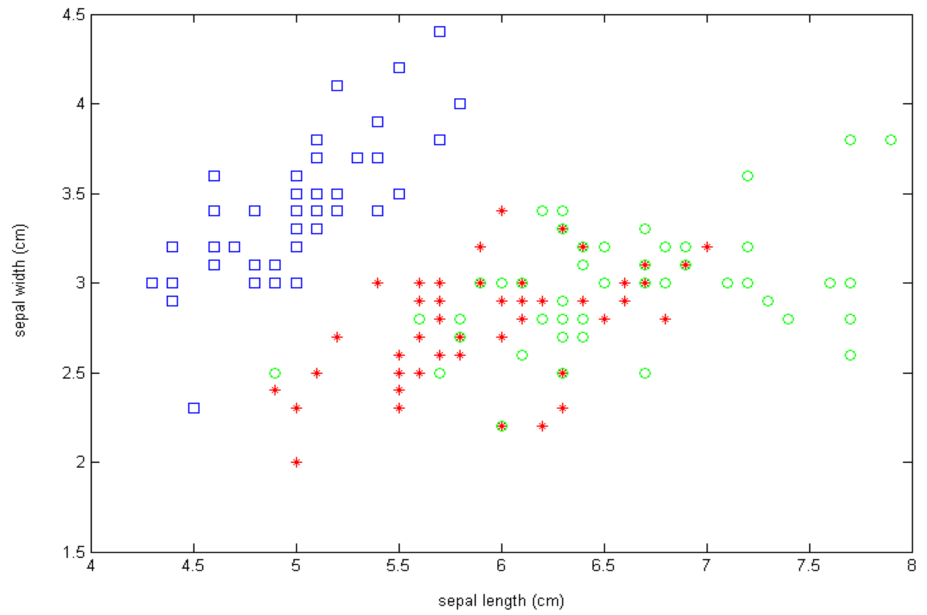


Figura 2.9 Representación de la muestra de Iris datasets haciendo uso de la técnica de visualización: Scatter plots (IRIS DATASET, 1996).

En la Figura 2.8 podemos visualizar un ejemplo del diagrama de dispersión en 3D, vemos como las diferentes variables son combinadas para formar coordenadas en espacio de fase.

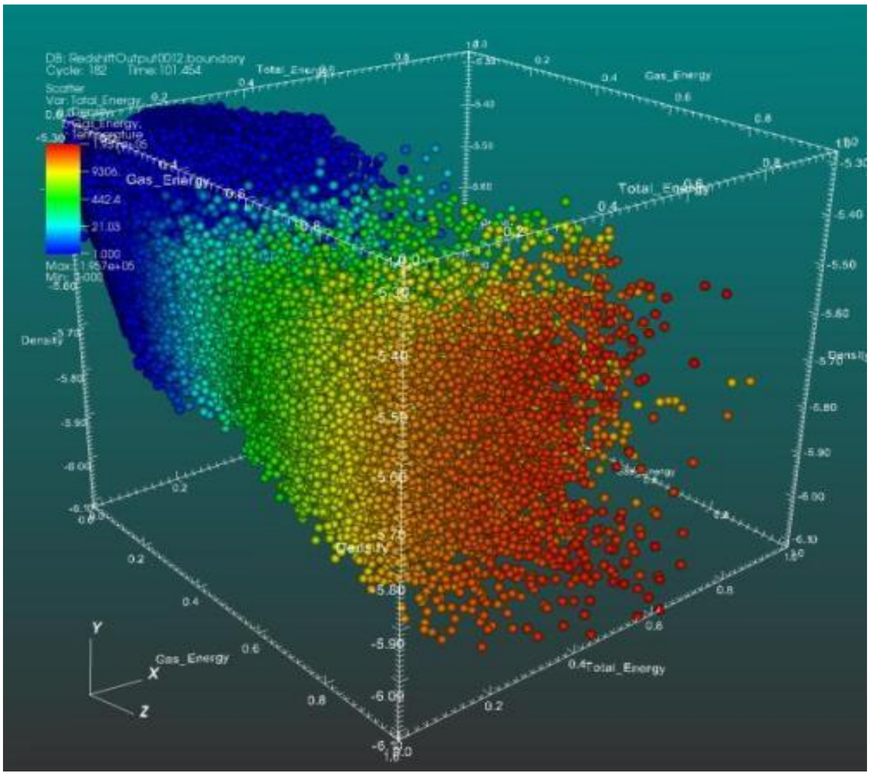


***Figura 2.10 Visualizaciones que han sido creadas con VisIt en wci.llnl.gov. última versión: Noviembre 8, 2007*[47]**

- Pair plots (scatter plot matrix-SPM): Un gráfico de matriz de dispersión es una herramienta de exploración de datos que permite comparar varios datasets para buscar patrones y relaciones. El gráfico de matriz de dispersión (SPM) acepta una capa o una tabla como entrada. Seleccione los campos que desee utilizar en el gráfico. El gráfico tiene dos componentes principales: una matriz de gráficos de dispersión pequeños para cada uno de los campos y una ventana Vista previa mayor que muestra el gráfico de dispersión para un par de campos seleccionados con mayor detalle. También puede habilitar el trazado de histogramas, mostrando la distribución de valores para cada uno de los campos. Una de sus desventajas principales como una técnica de visualización es su redundancia y la cantidad de espacio no usado (debido a la simetría). Pair plots últimamente pueden mostrar hasta una docena de variables, aunque el usuario pueda tener problemas visualmente procesando tantos datos.

[46] En probabilidad y estadística, la correlación indica la fuerza y la dirección de una relación lineal y proporcionalidad entre dos variables estadísticas.
[47] https://wci.llnl.gov/codes/visit/gallery.html

En la Figura 2.11 podemos observar la técnica de visualización de la matriz de dispersión - SPM, representando todas las dimensiones (sepal longitud, sepal anchura, petal longitud y petal anchura) del bien conocido conjunto de datos Iris (IRIS DATASET).

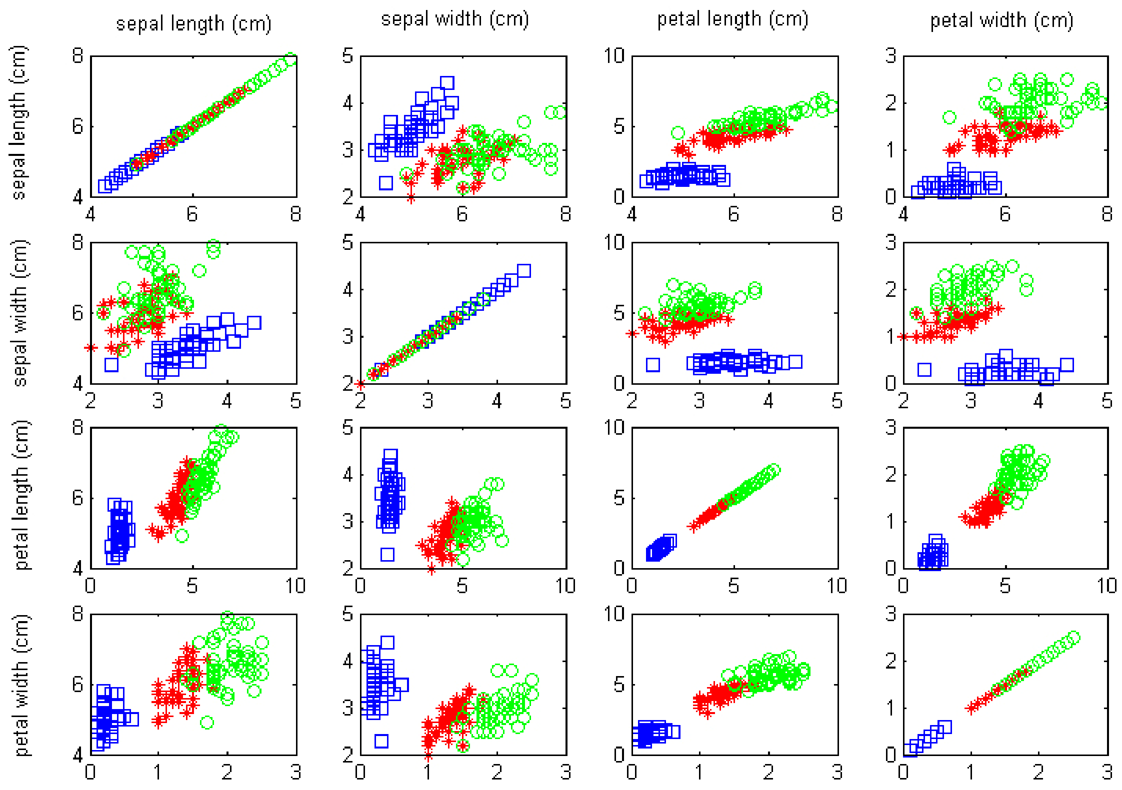


Figura 2.11 Representación de la muestra de Iris datasets (Pair Plots) (IRIS DATASET, 1996).

El hecho de que el espacio en el que nos movemos sea tridimensional (al menos aparentemente) hace difícil la representación visual de sistemas multidimensionales. Ello es tanto cuanto más difícil cuanto mayor es el número de dimensiones. La representación cartesiana resulta muy fácil de aprehender ya que visualmente coincide con aquello que vemos en nuestro mundo 3D. Una escapatoria a este límite en cuanto a dimensiones consiste en utilizar variables visuales, como el color, la orientación, la forma de cada elemento, etc. o añadir iconos a cada punto con el objetivo de añadir información. En cualquier caso siempre se mantiene la familiar referencia 2D o 3D.

Con este tipo de esquema se puede llegar a representar alrededor de una decena de dimensiones como mucho. Sin embargo, gran cantidad de problemas tienen una dimensionalidad netamente superior a la decena. La superación de este límite, fuerza a romper con el esquema cartesiano tradicional y nos lleva a buscar metáforas visuales desligadas del concepto habitual de espacio 2D o 3D. (DURSTELER, 2010)

En la Figura 2.12 se muestra como las coordenadas paralelas representan el conjunto de datos de Iris, el cual incluye todas sus dimensiones (sepal longitud, sepal anchura, petal longitud y petal anchura).

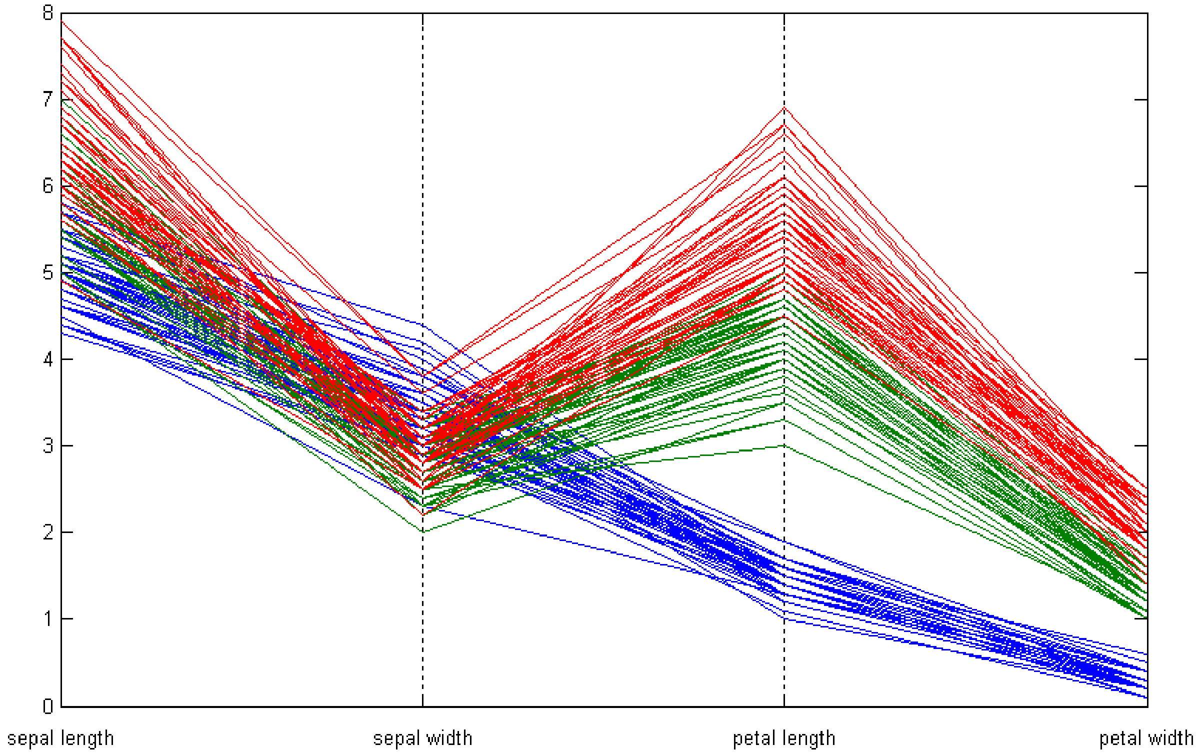


Figura 2.12 Representación de la muestra de Iris datasets (Parallel Coordinate Plots) (IRIS DATASET, 1996).

En la Figura 2.13 vemos un ejemplo de como las coordenadas paralelas son de gran utilidad en el Proyecto Sity'O'Scope**[48]**, que es un sistema de visualización interactiva de los datos recabados durante el año 2000 por el Swiss Economic Research Group of UBS (Grupo de investigación económica de la unión de bancos suizos) sobre las condiciones de vida, el coste y las ganancias de 58 ciudades en todo el mundo. Entre los datos recogidos se cuentan, por ejemplo, el tiempo que se necesita para ir a trabajar, el número de horas trabajadas al año, los salarios de distintas categorías profesionales, el coste de la vivienda o el tiempo de trabajo necesario para costearse un determinado bien de consumo. (DURSTELER, 2001)

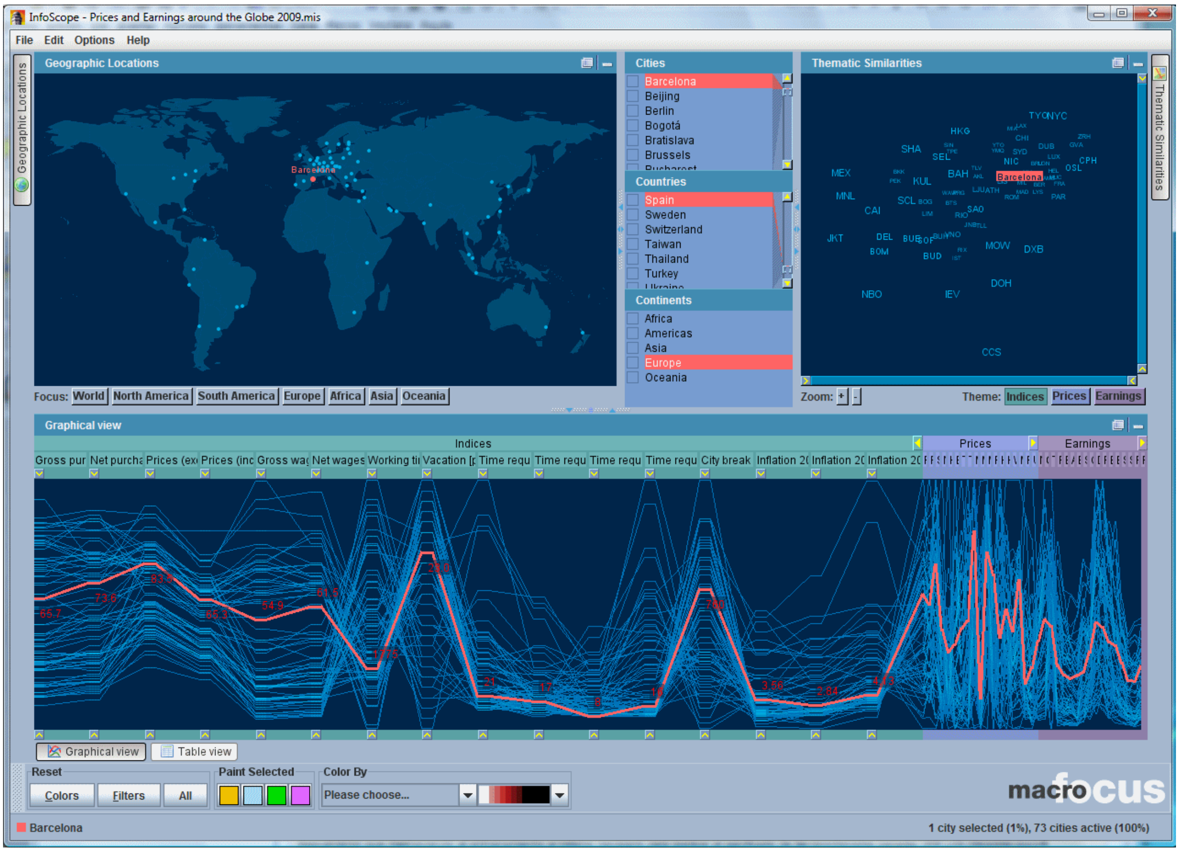


***Figura 2.13 Aspecto general de la ventana Sity'O'Scope*[48]**

[48] http://www.infovis.net/printMag.php?num=54&lang=1

3. CAPÍTULO TERCERO

"The ultimate security is your understanding of reality"

Howard Stanley Judd (New York, 1936)

Desde su inicio, el estudio de Redes neuronales Artificiales (SHELLY, 2010) ha ido creciendo en la conciencia que el cerebro humano calcula de un modo completamente diferente del computador digital convencional. La operación relativamente lenta de neuronas humanas (células nerviosas) es vencida por el número realmente asombroso de neuronas en el cerebro humano; cada uno con un número masivo de interconexiones. Los estudios han estimado que hay alrededor de 10 Billones de neuronas en la corteza humana, y los 60 trillones de conexiones-synapses-entre ellos, que dan a un una idea de su estructura sumamente compleja.

Así como de ser sumamente complejo, el cerebro es también un sistema no lineal, de información y procesamiento paralelos. Las neuronas son organizadas dentro del cerebro para realizar ciertos cómputos (p.ej. el modelo de reconocimiento, la percepción, y el control motor) mucho más rápido que una súper computadora. Como un ejemplo, se sabe que el cerebro rutinariamente logra tareas de reconocimiento perceptual (p.ej. el reconocer una cara antes sabida integrada en un entorno desconocido) en la orden de 100-200 ms, mientras una súper computadora tomará mucho más tiempo para realizar la misma tarea (si de verdad esto pudiera) (HERRERO, CORCHADO, 2013).

Los primeros modelos de redes neuronales datan de 1943 por los neurólogos McCulloch y Pitts. Años más tarde, en 1949, Donald Hebb desarrolló sus ideas sobre el aprendizaje neuronal, quedando reflejado en la "regla de Hebb". En 1958, Rosemblatt desarrolló el perceptrón**[49]** simple, y en 1960, Widrow y Hoff desarrollaron el ADALINE**[50]**, que fue la primera aplicación industrial real. En los años siguientes, se redujo la investigación, debido a la falta de modelos de aprendizaje y el estudio de Minsky y Papert sobre las limitaciones del perceptrón**[49]**. Sin embargo, en los años 80, volvieron a resurgir las RNA gracias al desarrollo de la red de Hopfield, y en especial, al algoritmo de aprendizaje de retropropagación ideado por Rumelhart y McLellan en 1986 que fue aplicado en el desarrollo de los perceptrones multicapa.**[51]**

Una primera clasificación de los modelos de redes neuronales es: Modelos inspirados en Biología, estos comprenden las redes que tratan de simular los sistemas neuronales Biológicos, así como ciertas funciones como las auditivas o de

[49] El Perceptrón es un tipo de red neuronal artificial desarrollado por Frank Rosenblatt, también puede entenderse como perceptrón la neurona artificial y unidad básica de inferencia en forma de discriminador lineal, que constituye este modelo de red neuronal artificial, esto debido a que el perceptrón puede usarse como neurona dentro de un perceptrón más grande u otro tipo de red neuronal artificial.
[50] El adaline (de ADAptative LINear Element) es un tipo de red neuronal artificial desarrollada por el profesor Bernie Widrow y su alumno Ted Hoff en la Universidad de Stanford en 1960.
[51]El perceptrón multicapa es una red neuronal artificial (RNA) formada por múltiples capas, esto le permite resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón (también llamado perceptrón simple). El perceptrón multicapa puede ser totalmente o localmente conectado. En el primer caso cada salida de una neurona de la capa "i" es entrada de todas las neuronas de la capa "i+1", mientras que en el segundo cada neurona de la capa "i" es entrada de una serie de neuronas (región) de la capa "i+1".

visión; y los Modelos Artificiales aplicados, estos modelos no tienen por qué guardar similitud estricta con los sistemas biológicos. Sus arquitecturas están bastante ligadas a las necesidades de las aplicaciones para las que son diseñadas.

Cada una de las áreas del conocimiento descritas anteriormente presentan en mayor o menor medida ciertas cuestiones aún no resueltas y sobre las cuales la comunidad científica plantea sus líneas de investigación, con mayor o menor intensidad; y de los cuales son el punto de partida para el desarrollo de la presente memoria de fin de máster. En el presente capítulo nos centraremos en el estudio de las redes neuronales, modelos, en los algoritmos de aprendizaje para la automatización de los pesos en las redes neuronales y aplicaciones de las redes neuronales para posteriormente representar todo el conocimiento relativo a los perfiles de comportamiento de los usuarios de una forma implícita al mecanismo de inferencia y no mediante los habituales esquemas explícitos inherentes a las aproximaciones basadas en perfiles, métricas estadísticas o sistemas expertos. De esta forma, se enriquece, potencialmente, el conocimiento relativo a las múltiples relaciones entre los diferentes parámetros de detección, con lo que la calidad de las decisiones inferidas es teóricamente superior (GARCÍA, 2006). Para ello organizaremos un esquema de funcionamiento compuesto por dos fases: aprendizaje, o entrenamiento, y explotación.

3.1 REDES NEURONALES ARTIFICIALES (ANN)

Redes neuronales artificiales es un concepto computacional que tiene por objetivo procesar informaciones de forma semejante al cerebro humano, el cual posee una gran capacidad de organizar y procesar informaciones de forma paralela, debido a la estructura de sus neurona.**[52]**

> *"... El cerebro és un computador (un sistema de procesamiento de la información) altamente complejo, no lineal y paralelo. Él tiene la capacidad de organizar sus representaciones estructurales, conocidas como neuronas, de la manera de realizar ciertos procesamientos mucho más rápido que la computadora digital más rápida que hoy en dia existe" (HAYKIN, 2001)."*

Para Lippmann apud (FERNANDES, 2005), las ANN son sistemas físicos que pueden adquirir, almacenar y utilizar conocimientos y alcanzar un buen rendimiento, debido a su interconexión entre los nodos de una red (arquitectura**[53]**). Lo que podemos determinar como más importante en una ANN, es su habilidad de aprendizaje **[54]** (FERNANDES, 2005).

3.1.1 INSPIRACIÓN BIOLÓGICA

El cerebro humano es formado por complejos circuitos neuronales cerebrales ampliamente interconectados, que son compuestos por células nerviosas, más cono-

[52] Neurona, es la unidad computacional básica de la red.
[53] Arquitectura, es la estructura topológica de la conexión de la neurona.
[54] Aprendizaje, es el proceso que adapta la red de modo de computar una función deseada

cidas como neurona biológica (ALMEIDA, 2000). Las neuronas y las conexiones entre ellas (sinapsis) constituyen la clave para el procesado de la información.

La mayor parte de las neuronas poseen una estructura de árbol, llamada dendritas, que reciben las señales de entrada procedentes de otras neuronas a través de la sinapsis. Una neurona consta de tres partes:

- El cuerpo de la neurona.
- Las dendritas, que reciben las entradas.
- El axón, que lleva la salida de la neurona a las dendritas de otras neuronas.

La forma completa en la que dos neuronas se relacionan no es totalmente conocida, y depende además del tipo particular de cada neurona. En general, una neurona envía su salida a otras por su axón, y este lleva la información por medio de diferencias de potencial eléctrico.

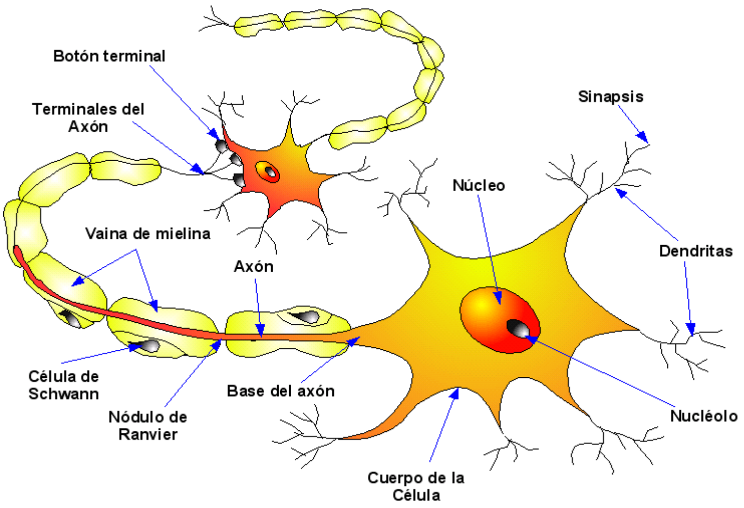


Figura 3.1 Neurona Biológica (HERRERO, CORCHADO, 2013).

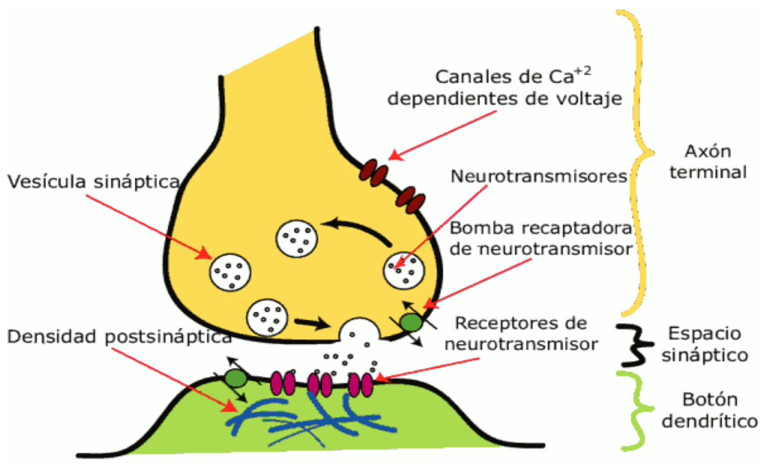


Figura 3.2 Sinapsis (ALMEIDA, 2000).

3.1.2 NEURONA ARTIFICIAL

La neurona artificial es el elemento básico de la estructura de una red neuronal artificial, conocido como nudo o elemento procesador, y fue proyectado basado en el funcionamiento de una neurona natural.

> *"La neurona artificial es una estrutura logico/matemática que intenta simular la forma, el comportamiento y las funciones de una neurona biológica. De esta manera, las dendritas son sustituidas por entradas, cuyas conexiones con el cuerpo celular artificial son realizadas a través de elementos llamados pesos (simulando las sinapsis). Los estímulos captados por las entradas son procesados por la función de suma, y el umbral de salida de la neurona biológica fue sustituida por la función de transferencia" (TAFNER, 1998).*

Cada neurona recibe un valor de entrada, el cual transforma según una función específica denominada función de activación. Dicha señal transformada pasa a ser la salida de la neurona. Las neuronas se conectan entre si según una determinada arquitectura. Cada conexión tiene un determinado peso que pondera cada entrada a la neurona. De esta manera la entrada de cada neurona es la suma de las salidas de las neuronas conectadas a ella, multiplicadas por el peso de la respectiva conexión. La figura siguiente ilustra dicho concepto:

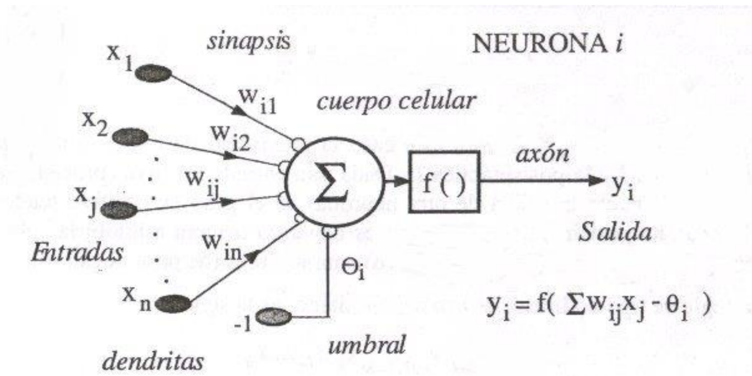


Figura 3.3 Esquema de funcionamiento de una neurona (TAFNER, 1998).

En este modelo, la neurona j recibe una serie de entradas x1, x2, …, xn. Cada señal se multiplica por el peso asociado a su conexión, w1, w2,…,wn. Luego, se suman estas entradas ponderadas y se les aplica la función de activación F(.) para generar la señal de salida de la neurona j. Los valores de los pesos son ajustados durante la fase de aprendizaje.

Como se ha comentado anteriormente, estas neuronas están conectadas entre sí de acuerdo a una determinada arquitectura. Es decir, las neuronas se agrupan en distintas capas: *una capa de entrada, otra de salida*, y en el caso de existir, una o varias *capas ocultas*. La salida de cada neurona se propaga por igual por estas conexiones hasta las neuronas de destino. Cada conexión tiene un peso asociado que pondera el valor numérico de la señal que viaja por esta. Así pues, una red de neuronas artificial puede verse como un grafo cuyos nodos tienen funcionamiento similar, los cuales propagan la información a través de las distintas conexiones.

Una red neuronal artificial consiste en varias neuronas organizadas según una cierta topología de red. Un ejemplo de una topología de red que consiste en una capa de entrada, dos capas ocultas y una capa de salida son mostradas en la Figura 3.4

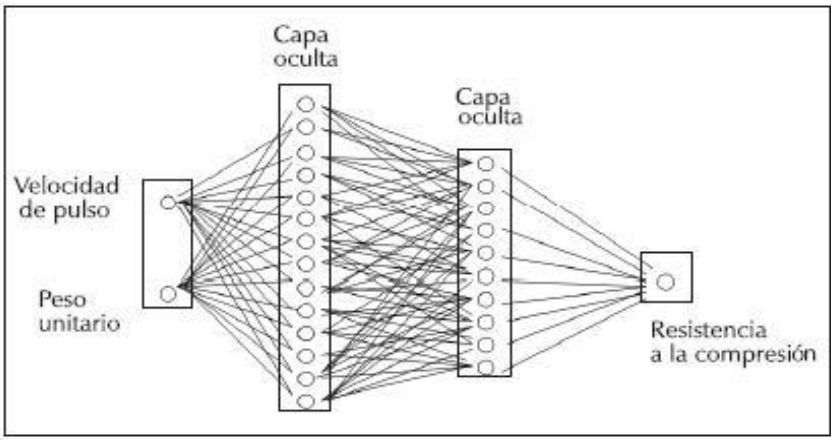


Figura 3.4 Arquitectura de la red neuronal.
Fuente: http://www.scielo.org.co/pdf/iei/v27n1/v27n1a02.pdf

La Figura 3.5 muestra una red neuronal multinivel utilizada para el reconocimiento óptico de caracteres. Cuenta con 35 neuronas en la capa de entrada, 10 neuronas en la capa oculta y 27 neuronas en la capa de salida. El número de neuronas en la capa de entrada obedece a la longitud de los patrones de entrenamiento y el número de neuronas en la capa de salida es dado por las 27 letras del alfabeto. Las diez neuronas de la capa oculta son seleccionadas después de experimentar con diferentes cantidades y obtener mejores resultados con ese número.

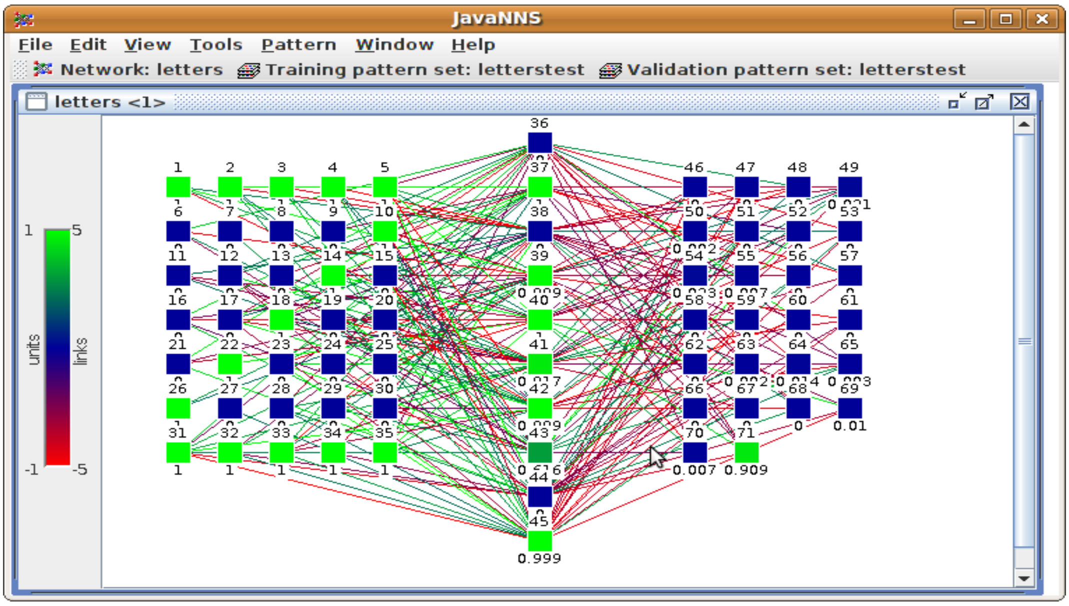


Figura 3.5 Simulador JavaNNS desarrollado en la Universidad de Tübingen. Ejemplo de una red neuronal artificial con una capa de entrada, una capa oculta y una capa de salida.

3.1.3 APRENDIZAJE AUTOMATICO (MACHINE LEARNING)

El Aprendizaje Automático o Aprendizaje de Máquinas es una rama de la Inteligencia Artificial[55] cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender. De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos. Es, por lo tanto, un proceso de inducción del conocimiento. En muchas ocasiones el campo de actuación del Aprendizaje Automático se solapa con el de la Estadística, ya que las dos disciplinas se basan en el análisis de datos. Sin embargo, el Aprendizaje Automático se centra más en el estudio de la Complejidad Computacional[56] de los problemas. Muchos problemas son de clase NP-hard[57], por lo que gran parte de la investigación realizada en Aprendizaje Automático está enfocada al diseño de soluciones factibles a esos problemas. El Aprendizaje Automático puede ser visto como un intento de automatizar algunas partes del Método Científico[58] mediante métodos matemáticos.

El Aprendizaje Automático tiene una amplia gama de aplicaciones, incluyendo motores de búsqueda, diagnósticos médicos, detección de fraude en el uso de tarjetas de crédito, análisis del mercado de valores, clasificación de secuencias de ADN, reconocimiento del habla y del lenguaje escrito, juegos y robótica.

3.1.4 ALGORITMOS DE APRENDIZAJE (LEARNING ALGORITHMS)

Los diferentes algoritmos de aprendizaje automático se agrupan en una taxonomía en función de la salida de los mismos. Algunos algoritmos son:

3.1.5 APRENDIZAJE SUPERVISADO (SUPERVISED LEARNING)

En aprendizaje automático[59] y minería de datos[60], el aprendizaje supervisado es una técnica para deducir una función a partir de datos de entrenamiento. Los datos de entrenamiento consisten de pares de objetos (normalmente vectores): una componente del par son los datos de entrada y el otro, los resultados deseados. La salida de la función puede ser un valor numérico (como en los problemas de regresión[61]) o una etiqueta de clase (como en los de clasificación).

---

[55] En ciencias de la computación se denomina inteligencia artificial (IA) a la capacidad de razonar de un agente no vivo. (CSAIL, DECSAI, STANFORD)

[56] La Teoría de la Complejidad Computacional es una rama de la teoría de la computación que se centra en la clasificación de los problemas computacionales de acuerdo a su dificultad inherente, y en la relación entre dichas clases de complejidad.

[57] En teoría de la complejidad computacional, la clase de complejidad NP-hard (o NP-complejo, o NP-difícil) es el conjunto de los problemas de decisión que contiene los problemas H tales que todo problema L en NP puede ser transformado polinomialmente en H. Esta clase puede ser descrita como conteniendo los problemas de decisión que son al menos tan difíciles como un problema de NP.

[58] Es un método de investigación usado principalmente en la producción de conocimiento en las ciencias.

[59] Véase el apartado "3.1.3".

[60] La minería de datos (es la etapa de análisis de "Knowledge Discovery in Databases" o KDD), es un campo de las ciencias de la computación referido al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos. Utiliza los métodos de la inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos.

[61] las pruebas de regresión son cualquier tipo de pruebas de software que intentan descubrir las causas de nuevos errores (bugs), carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, inducidos por cambios recientemente realizados en partes de la aplicación que anteriormente al citado cambio no eran propensas a este tipo de error.

Este tipo de aprendizaje se caracteriza por la presencia de un “Profesor” o supervisor, externo a la red, quien posee el conocimiento sobre el ambiente. El Conjunto de entradas y las respectivas salidas representan el conocimiento citado.

El supervisor posee la función de repasar su conocimiento del ambiente a la red, además de monitorear la respuesta de salida. El repasa su conocimiento suministrando una respuesta deseada para cada entrada realizada. Esa respuesta representa la acción óptima que debe ser realizada por la red neuronal o ajuste de pesos es realizado bajo la influencia combinada del conjunto de entrenamiento y de la señal de error. Entiéndase como señal de error a la distancia entre la respuesta actual de la red y la deseada. (FERNANDES, 2005)

Para realizar el aprendizaje, la red necesita poseer un conjunto de entradas y un conjunto con las salidas deseadas para cada entrada. Toda vez que se va presentar una entrada, deberá ser verificada la salida obtenida y verificar si esa salida verifica con la entrada. La salida obtenida es generada a través de cálculos realizados con los pesos que la red posee. Es necesario realizar ese tipo de entrenamiento con todo el conjunto de entradas y salidas hasta que la taza de acierto sea considerada satisfactoria.

Con la realización repetitiva de entrenamiento, todo conocimiento disponible por el supervisor es repasado por la red neuronal artificial. Cuando la transferencia se realiza por completo, podemos dejar de lado la presencia del supervisor y dejar a la red neuronal lidiar con las nuevas situaciones por sí misma.

En la Figura 3.6 se muestra el aprendizaje supervisado.

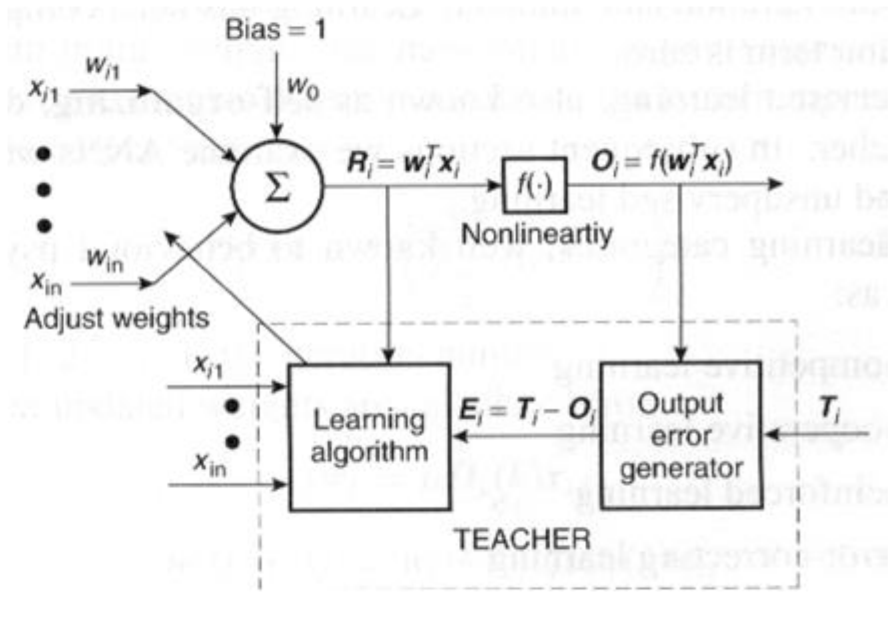


Figura 3.6 Aprendizaje supervisado (FERNANDES, 2005)

3.1.6 APRENDIZAJE NO SUPERVISADO (UNSUPERVISED LEARNING)

Aprendizaje no supervisado es un método de Aprendizaje Automático[62] donde un modelo es ajustado a las observaciones. Se distingue del Aprendizaje supervisado por el hecho de que no hay un conocimiento a priori. En el aprendizaje no supervisado, un conjunto de datos de objetos de entrada es tratado. Así, el aprendizaje no supervisado típicamente trata los objetos de entrada como un conjunto de variables aleatorias[63], siendo construido un modelo de densidad para el conjunto de datos.

El aprendizaje no supervisado puede ser usado en conjunto con la Inferencia bayesiana[64] para producir probabilidades condicionales (es decir, aprendizaje supervisado) para cualquiera de las variables aleatorias dadas. El Santo Grial del aprendizaje no supervisado es la creación de un código factorial de los datos, esto es, un código con componentes estadísticamente independientes. El aprendizaje supervisado normalmente funciona mucho mejor cuando los datos iniciales son primero traducidos en un código factorial. El aprendizaje no supervisado también es útil para la compresión de datos: fundamentalmente, todos los algoritmos de compresión dependen tanto explícita como implícitamente de una distribución de probabilidad sobre un conjunto de entrada.

> *“La capa de entrada recibe los datos disponibles. La capa competitiva consiste de neuronas que compiten entre sí (de acuerdo con una regla de aprendizaje) por la “oportunidad” de responder a las características contenidas en los datos de entrada. En su forma más simple, la red opera de acuerdo con una estrategia del tipo “el vencedor lleva todo.” En esta estrategia la neurona con la mayor entrada total “gana” la competición y se conecta; todos las otras neuronas, entonces se desconectan.”(HAYKIN, 2001)*

Las redes del tipo de aprendizaje no supervisado son utilizadas como detector de características. En la Figura 3.7 es posible visualizar este tipo de aprendizaje.

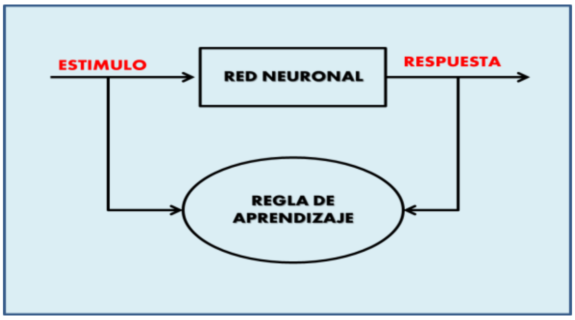


Figura 3.7 Aprendizaje no supervisado

[62] Véase el apartado 3.1.3
[63] En probabilidad y estadística, una variable aleatoria o variable estocástica es una variable estadística cuyos valores se obtienen de mediciones en algún tipo de experimento aleatorio.
[64] La inferencia bayesiana es un tipo de inferencia estadística en la que las evidencias u observaciones se emplean para actualizar o inferir la probabilidad de que una hipótesis pueda ser cierta.

3.1.6.1 APRENDIZAJE HEBBIANO (HEBBIAN LEARNING)

La teoría Hebbiana describe un mecanismo básico de plasticidad sináptica**[65]** en el que el valor de una conexión sináptica se incrementa si las neuronas de ambos lados de dicha sinapsis se activan repetidas veces de forma simultánea. Introducida por Donald Hebb, en 1949, es también llamada regla de Hebb, postulado de aprendizaje de Hebb o Teoría de la Asamblea Celular, y afirma lo siguiente:

> *"When an axon of a cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency as one of the cells firing B, is in-creased" .(HEBB, 1949)*
>
> *“Cuando un axón de una célula A está lo suficientemente cerca de una célula B, como para excitarla, y repetidas veces toma parte en dicha activación, algunos procesos de crecimiento o cambios metabólicos tienen lugar en una o ambas células tal que la eficiencia de A como una de las células que activan a la célula B, aumenta.”*

Esta declaración a veces es ampliada (HAYKIN, 1994) en una regla de dos partes:

1. Si dos neuronas de todos lados de un sinapsis son activadas simultáneamente (p. ej. sincrónicamente), entonces el peso sináptico que las une aumenta.

2. Si dos neuronas de todos lados de un sinapsis son activadas asincrónicamente, entonces el peso sináptico es debilitada o eliminada.

Considerando una red neuronal básica anticipativa, esto sería interpretado de manera siguiente: el peso entre una neurona de entrada y una neurona de salida enormemente es reforzado cuando la activación de la neurona de entrada hace que la neurona de salida se active fuertemente. Puede ser visto que la regla favorece al fuerte: si los pesos entre entradas y salidas son ya grandes (y entonces una entrada tendrá un efecto fuerte sobre las salidas), es probable que los pesos crecerán. En términos matemáticos, si consideramos la red neuronal anticipativa más simple que tiene una capa de entrada asociadas a un vector de entrada, x, y una capa de salida asociadas a un vector de salidas, y, entonces tenemos la siguiente ecuación (HERRERO, CORCHADO, 2013):

$$y_i = \sum_j W_{ij} X_j \qquad \textbf{(3.1)}$$

Donde W_{ij} representa el vector peso entre la entrada “y” la salida “i”. La regla de aprendizaje Hebbian es definido por:

---

[65] La plasticidad neuronal, también denominada neuroplasticidad, plasticidad neural o plasticidad sináptica, es la propiedad que emerge de la naturaleza y funcionamiento de las neuronas cuando éstas establecen comunicación, y que modula la percepción de los estímulos con el medio, tanto los que entran como los que salen.

$$\Delta W_{ij} = nX_j y_j \qquad \textbf{(3.2)}$$

Donde n es el coeficiente de aprendizaje. Esto quiere decir que el peso entre la neurona de entrada y la de salida es aumentado en la proporción a la magnitud del simultáneo de estas neuronas. El valor calculado puede ser sustituido en la regla de aprendizaje por:

Las propiedades estadísticas de las reglas de aprendizaje son acentuados en la expresión pasada. Esto puede ser visto como la regla de aprendizaje depende sobre la correlación entre las partes diferentes de los datos de entradas de los componentes del vector.

El valor de "y" calculado por la activación directa de la actividad, puede ser sustituida en la regla de aprendizaje siguiente:

$$\Delta W_{ij} = nX_j \sum_j W_{ik} X_k = n \sum_k W_{ik} X_k X_j \qquad \textbf{(3.3)}$$

Las propiedades estadísticas de la regla de aprendizaje son acentuadas en esta última expresión. Esto puede ser visto como la regla de aprendizaje depende de la correlación entre las diferentes partes de los datos de entrada de los componentes del vector.

Frecuentemente, la evaluación final de los pesos es la parte más interesante después de entrenar la red. Esto permitirá la predicción de como la red reaccionará a nuevas entradas. Sin embargo, esto es sólo posible si los pesos convergen a conjuntos específicos de valores. (HERRERO, CORCHADO, 2013)

La ley de Hebb no es la única regla de aprendizaje. Existe una amplia diversidad de algoritmos de aprendizaje. Haykin considera cinco tipos básicos de reglas de aprendizaje (HAYKIN, 1994):

- Aprendizaje basado en la corrección del error.
- Aprendizaje basado en la memoria.
- Aprendizaje Hebbiano.
- Aprendizaje competitivo.
- Aprendizaje de Boltzmann.

La mayor parte de los métodos neuronales hablados en la presente memoria están basados en el estudio de Hebbian.

- **APRENDIZAJE ANTI-HEBBIANO - ANTI-HEBBIAN LEARNING**

Si correlacionamos la activación de neuronas en una red neuronal, entonces

cada uno contiene la información sobre el otro. Así hay redundancia de información en las activaciones. El aprendizaje Anti-Hebbian es usado decorrelacionar las respuestas de neurona. De este modo, más información puede ser pasada a una red cuando los nodos de la red son todos tratados con datos diferentes. El objetivo es de conseguir que las neuronas que respondan a diferentes estímulos. La regla de aprendizaje El anti-Hebbian es representada por:

$$\Delta W_{ij} = n y_i y_j \quad \textbf{(3.4)}$$

Esto quiere decir que si dos salidas, “y_i” y “y_j”, son altamente correlacionadas entonces los pesos entre ellos crecerán a un gran valor negativo y cada uno tenderá a eliminar a la otra.

Ha sido mostrado (RUBNER,1989; RUBNER,1990) que la actualización de los pesos por el aprendizaje Anti-Hebbian convergerán hacia valores tal que la correspondencia “y_i” y “y_j” son decorrelacionadas. No hay ninguna necesidad de limitar el crecimiento de tales pesos como el proceso de aprendizaje Anti-Hebbian es intrínsecamente autorestrictivo.

- **ANALISIS DE COMPONENTES PRINCIPALES - PRINCIPAL COMPONENT ANALYSIS (PCA)**

El Análisis de Componentes Principales (PCA) es una técnica estadística de síntesis de la información, o reducción de la dimensionalidad (número de variables) (HOTELLING, 1933). Es decir, ante un conjunto de datos multivariables, el objetivo será reducirlas a un menor número perdiendo la menor cantidad de información posible. Los nuevos componentes principales o factores serán una combinación lineal de las variables originales, y además serán independientes entre sí. Un aspecto clave en PCA es la interpretación de los factores, ya que ésta no viene dada a priori, sino que será deducida tras observar la relación de los factores con las variables iniciales (habrá, pues, que estudiar tanto el signo como la magnitud de las correlaciones). Esto no siempre es fácil, y será de vital importancia el conocimiento que el experto tenga sobre la materia de investigación (TERRÁDEZ, UOC).

En el PCA existe la opción de usar la matriz de correlaciones o bien, la matriz de covarianzas. En la primera opción se le está dando la misma importancia a todas y a cada una de las variables; esto puede ser conveniente cuando el investigador considera que todas las variables son igualmente relevantes. La segunda opción se puede utilizar cuando todas las variables tengan las mismas unidades de medida y además, cuando el investigador juzga conveniente destacar cada una de las variables en función de su grado de variabilidad (GONZÁLEZ D, DÍAZ, TORRES, GARNICA, 2003).

Los únicos requerimientos previos para la aplicación del PCA son:

a) Continuidad en las variables.

b) El número “n” de individuos o elementos observados debe ser mayor que el número “p” de variables originales.

Esta reducción de la dimensión va a permitir:

a) Simplificar posteriores análisis, que se harán a partir de un menor número de variables que el original.

b) Una representación gráfica de los individuos en dimensión reducida.

c) Examinar e interpretar las relaciones entre las variables observadas.

Por otra parte, el PCA tiene la ventaja de no exigir supuestos tales como la normalidad u homoscedasticidad.

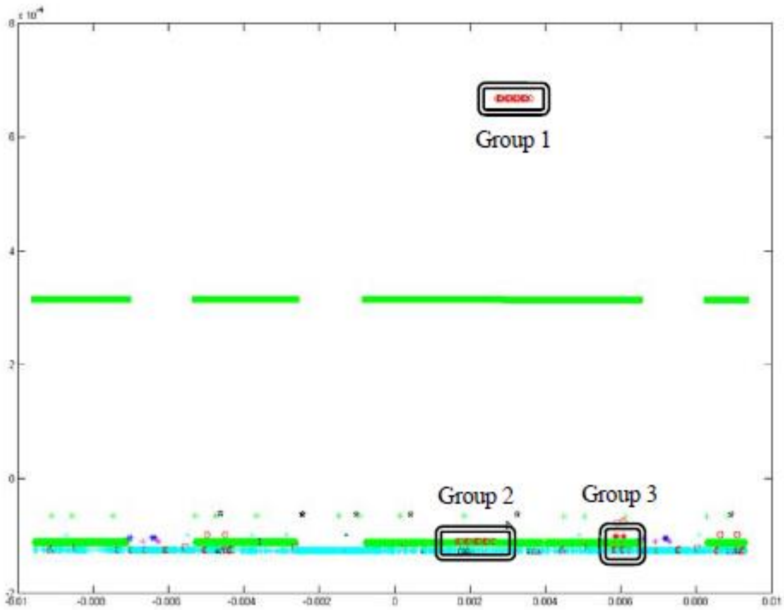


Figura 3.8 PCA, visualización de las pruebas realizadas en MOVICAB-IDS SYSTEM (CORCHADO, HERRERO, 2013).

En el libro “Mobile Hybrid Intrusion Detection: MOVICAB-IDS System”, publicado por los doctores Álvaro Herrero y Emilio Corchado encontramos una excelente explicación matemática del PCA, siendo detallado a continuación:

PCA puede ser descrito como el mapeo de vectores donde “x^d” en espacio N-dimensional en vectores de entrada (x_1, x_2, … x_n), “y^d” en un espacio M-dimensional en vectores de salida (y_1, y_2, … y_n), donde $M \leq N$. “x” puede ser re-

presentado como una combinación lineal de un conjunto de N vectores ortonormales "w_i" :

$$X = \sum_{i=1}^{N} Y_i \; W_i \tag{3.5}$$

Vectores "W_i" satisface la relación ortonormal.

$$W_i^t W_j = \delta_{ij} \tag{3.6}$$

Donde "δ_{ij}" es la Kronecker delta.
Haciendo uso de la ecuación (3.4), el coeficiente "y_j" podría ser dado por

$$y_i = W_i^t X \tag{3.7}$$

El cual puede ser considerado como una simple rotación del sistema de coordenada de los valores originales "x" a un nuevo conjunto de coordenadas dadas por "y". Si solamente un subconjunto M < N de la base de vectores, es conservado de modo que solamente M coeficientes "y" sean usados, y habiendo substituido los coeficientes restantes por constantes "b_i", entonces cada "x" vector podría ser aproximado por la expresión siguiente:

$$\tilde{X} = \sum_{i=1}^{M} y_i \, W_i + \sum_{i=M+1}^{N} b_i \; W_i \tag{3.8}$$

Considerando todo el conjunto de datos de "D" vectores, "x^d" donde d = 1,…,D. Esto es necesario para escoger la mejor opción de "W_i" y "b_i" de modo que "y_i" valores obtenidos por la ecuación (3.7) da la mejor aproximación de la ecuación (3.8) que todo el conjunto de datos. El Vector "x^d" tiene un error debido a la reducción de la dimensionalidad.

$$X^d - \tilde{X}^d = \sum_{i=M+1}^{N} \left(y_i^d - b_i\right) W_i \tag{3.9}$$

La mejor aproximación puede ser definida como la reducción al mínimo de la suma de los cuadrados de los errores sobre el conjunto total de datos:

$$E_M = \frac{1}{2}\sum_{d=1}^{D}\left\|X^d - \tilde{X}^d\right\|^2 = \frac{1}{2}\sum_{d=1}^{D}\sum_{i=M+1}^{N}(y_i^d - b_i)^2$$

(3.10)

Si nosotros calculamos la derivada de "E_M" con respecto a "b_i" y se establece a cero entonces:

$$b_i = \frac{1}{D}\sum_{d=1}^{D} y_i^d = W_i^T\,\bar{X}\ \forall i\ M+1, \dots, N$$

(3.11)

Donde el significado del vector de la media simple "*x*" es definido como:

$$\bar{X} = \frac{1}{D}\sum_{d=1}^{D} X^d$$

(3.12)

Ahora la suma de los cuadrados de los errores pueden ser definidas como:

$$E_M = \frac{1}{2}\sum_{i=M+1}^{N}\sum_{d=1}^{D}\{W_i^T(X^d - \bar{X})\}^2 = \frac{1}{2}\sum_{i=M+1}^{d} W_i^T \sum W_i$$

(3.13)

Donde la sumatoria es la matriz de covarianza de la muestra del conjunto de vectores $\{x^d\}$ y es dado por:

$$\sum = \frac{1}{D}\sum_{d}(X^d - \bar{X})(X^d - \bar{X})^T$$

(3.14)

Entonces si reducimos al mínimo E_M con respecto a la opción "W_i" esto puede ser mostrado (BISHOP, 1996) que el mínimo ocurre cuando la base de los vectores satisface:

$$\sum W_i = \lambda_i W_i$$

(3.15)

De ahí, aquellos "W_i" vectores propios de la matriz de covarianza. Tal cual es asumido, el vector propio puede ser probado para ser ortogonal, si la matriz de covarianza es real y simétrica. Si substituimos la ecuación (3.15) en la ecuación (3.13) y hacemos uso de la ortonormalidad en la ecuación (3.6), entonces el valor

del criterio de error en el mínimo puede ser representado en la forma siguiente:

$$E_M = \frac{1}{2} \sum_{i=M+1}^{N} \lambda_i \qquad (3.16)$$

Consecuentemente, el mínimo de error es obtenido por escoger el más pequeño valores de los valores propios (N-M), y su correspondiente vectores propios, como estos para ser desechados. Nosotros frecuentemente nos referimos a "y 's" como la componente principal.

- **MODELO DE OJA - OJA'S WEIGHTED SUBSPACE ALGORITHM**

El profesor Erkki Oja, a principios de los años 80, trata de resolver la inestabilidad del aprendizaje Hbiano originada por el crecimiento incontrolado de los pesos causado por la realimentación positiva. A continuación se detallan tres de sus modelos más populares:

➢ **MODELO DE UNA NEURONA DE OJA**

Erkki Oja desarrolló un modelo de aprendizaje Hebbiano con decremento de pesos que no sólo evita que los pesos crezcan sin control sino que también causa que los pesos converjan a las componentes principales de un conjunto de datos de entrada. Él sugirió una simple neurona de salida, la cual suma las entradas de la forma tradicional.

$$y = \sum_{i=1}^{N} W_i X_i \qquad (3.17)$$

Introdujo una variación para la regla de Hebb de la siguiente manera:

$$\Delta W_i = n(X_i\, y - y^2\, W_i) \qquad (3.18)$$

Se observa que el término de decremento de peso hace que "W_i" se aproxime a 1 y converge al principal vector propio de la matriz de covarianza de entrada. Los pesos están normalizados, por lo que no crecen sin control. Esta regla sólo encuentra el primer vector propio (la dirección correspondiente al mayor valor propio) de la matriz de covarianza de los datos. Para encontrar más componentes principales se necesita que haya interacción entre varias neuronas.

➢ EL ALGORITMO DEL SUBESPACIO DE OJA

Este algoritmo (Oja`S Subspace Algorithm) parte de una RNA con *N* neuronas en la capa de salida, cada una de las cuales aprende usando la regla de Hebb con un término de caída para que los pesos no crezcan ilimitadamente. Esto no garantiza el encontrar los componentes principales, pero si asegurar que los pesos converjan a una base ortogonal del espacio de los componentes principales.

Se llama al espacio expandido por esta base "Subespacio Principal". La regla de aprendizaje es como se expresa a continuación:

$$\Delta W_{ij} = n\,(X_i\, y_i - \; y_i \sum_{k} W_{kj} y_k)$$

(3.19)

Fuerza a los pesos a converger a una base del subespacio principal. Es un método homogéneo ya que las operaciones que afectan a cada neurona son las mismas. Tiene la desventaja de no encontrar los vectores propios, ya que sólo encuentra el subespacio principal.

➢ SUBESPACIO PESADO DE OJA

En este algoritmo (Oja`S Weighted Subspace Algorithm) se introdujo asimetría para limitar el valor de los pesos y así forzarlos a converger a los componentes principales. El modelo está definido con las siguientes ecuaciones:

$$y_i = \sum_{i=1}^{N} W_{ij} X_j$$

(3.20)

Donde una regla de tipo Hebb con término de caída modifica los pesos de acuerdo a:

$$\Delta W_{ij} = n y_i (X_j - \; \theta_i \sum_{K=1}^{M} W_{kj} y_k)$$

(3.21)

Asegurando que $\theta_1 < \theta_2 < \ldots$ permite a la neurona cuyo peso decrece proporcionalmente a aprender los valores principales de correlación en el conjunto de datos de entrada. Esto significa que esta primera neurona responderá a las direcciones paralelas a la primera componente principal. La segunda salida no puede competir con la primera, pero está en una posición más idónea para identificar la segunda componente principal, y así sucesivamente para todas las salidas en la red neuronal. Se puede mostrar que el vector de pesos convergerá a los vectores propios en el orden de sus valores propios.

- **RED CON REALIMENTACION NEGATIVA - NEGATIVE FEEDBACK NETWORK(NFN)**

La Red con Realimentación Negativa (Negative Feedback Network) permite realizar análisis de componentes principales, al igual que el Subespacio de Oja, permitiendo trabajar con el residuo. En el artículo científico desarrollado por (CORCHADO, 2010) vemos el diseño, la evaluación e implementación de la red de alimentación negativa para nVidia Compute Unified Device Architecture (CUDA) platform .

Esta RNA es capaz de encontrar las componentes principales de los datos de entrada de una forma equivalente al algoritmo del Subespacio de Oja. Por tanto, los pesos no encontrarán las componentes principales pero si una base del subespacio principal.(CORCHADO, HERRERO, 2013)

- **ANALISIS FACTORIAL(AF)**

El Análisis Factorial (AF) es una técnica que persigue resumir la información contenida en un conjunto de datos identificando un reducido número de factores, siendo el número de este menor que el número de variables originales, de manera que cada variable se pueda expresar como una combinación lineal de dichos factores, los cuáles no son directamente observables.

El origen de AF se atribuye a Spearman**[66]**. Éste, apoyado en esta técnica propuso la existencia de un factor general de inteligencia, que subyace a las habilidades para la ejecución de las tareas intelectuales. AF es usado en campos como la psicología y las ciencias sociales, la economía, entre otros.

- **EXPLORATORY PROJECTION PURSUIT (EPP)**

La Búsqueda Exploratoria de Proyecciones (Exploratory Proyection Pursuit) (EPP) es un método estadístico para resolver problemas en los que es difícil encontrar estructura interna en datos de alta dimensionalidad. Se debe definir un índice que mida cómo es de importante una proyección. Después, los datos se transforman maximizando dicho índice con el objetivo de maximizar el interés de acuerdo con él.

Una proyección aleatoria típica es Gaussiana, y estadísticamente, las proyecciones más interesantes son las que se alejan de dicha distribución.

Las distribuciones Gaussianas están totalmente definidas por sus dos primeros momentos, la media y la varianza. Los dos momentos siguientes son el coeficiente de asimetría o desplazamiento (skewness) y la curtosis. Normalmente los datos se “blanquean” (“sphering”) obteniendo una media cero y una matriz de

[66] Charles Edward Spearman (Londres, 10 de septiembre de 1863 - Londres, 7 septiembre de 1945). Psicólogo inglés. Estudió en las universidades de Leipzig, Wurzburgo y Göttingen y enseñó e investigó en la Universidad de Londres (1907 - 1931). Formuló la teoría de que la inteligencia se compone de un factor general y otros específicos. Creyó en la existencia de un factor general que interviene en todas las fases de la conducta humana y atribuyó a las capacidades específicas papel determinante en cada actividad. Escribió *The Abilities of Man* (1927), *Creative Mind* (1930) y *Psychology Down the Ages* (1937).

covarianza identidad forzando a que la RNA responda a órdenes estadísticos mayores y pueda encontrar estructura en los datos.

El coeficiente de asimetría se basa en el tercer momento normalizado de una distribución y se refiere a cómo de asimétrica es una distribución, expresado en la ecuación (22). La curtosis se basa en el cuarto orden de normalización de la distribución e indica cómo es de apuntada una distribución, definida en la ecuación (23).

$$g(s) = \frac{E((s-\mu)^3)}{\sigma^3} \qquad \textbf{(3.22)}$$

$$g(s) = \frac{E((s-\mu)^4)}{\sigma^4} \qquad \textbf{(3.23)}$$

Donde S es una variable aleatoria desde la distribución con media μ y desviación estándar σ. Hay tres tipos diferentes de distribuciones en base a la curtosis. Una curva normal es mesocúrtica y presenta un grado de observaciones medio en la zona central. Las distribuciones platicúrticas poseen menos observaciones en los extremos y más en el centro; y las leptocúrticas, son de forma puntiaguda, con menos observaciones en la zona central de la distribución y másen los extremos. En la Figura 3.9 podemos observar los tres tipos de distribuciones en base a la curtosis.

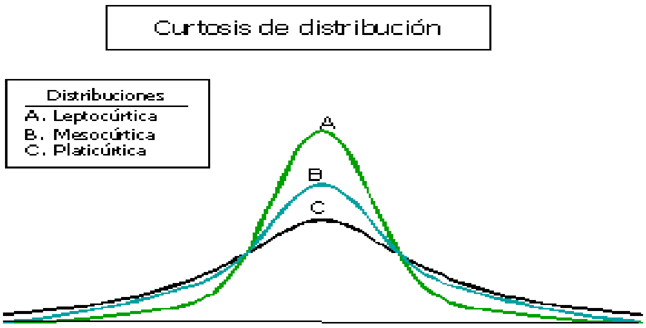


Figura 3.9 Curtosis de distribución

- **GENERALIZACIÓN DEL APRENDIZAJE HEBBIANO PARA LA MAXIMIZACIÓN DE LA PROBALIDAD DEL RESIDUO - MAXIMUN LIKELIHOOD HEBBIAN LEARNING (MLHL)**

Una implementación neuronal de EPP**[67]** es una generalización del aprendizaje Hebbiando para la maximización de la probabilidad del residuo (MLHL, Maximum Likelihood Hebbian Learning). Las reglas de aprendizaje de la red con realimentación negativa**[68]** pueden ser derivadas como una aproximación a la mejor compresión lineal de los datos, partiendo de la función de coste:

[67] Véase el apartado 3.1.10
[68] Véase el apartado 3.1.8

$$J = 1^T E\{(X - Wy)^2\}$$

(3.24)

Que se minimiza para conseguir la ecuación (29). La minimización de *"J"* es equivalente a minimizar la probabilidad negativa del logaritmo del residuo, *"e"*, por lo que si se tiene:

$$p(e) = \frac{1}{Z}\exp(-e^2)$$

(3.25)

Donde "p(e)" es la función de densidad que se corresponde con la probabilidad del residuo después de la retroalimentación negativa, se tendrá la siguiente función general de coste asociada a la red neuronal:

$$J = -\log p(e) = (e)^2 + k$$

(3.26)

Donde *"k"* es una constante. Aplicando el gradiente descendente en *"J"* se obtiene:

$$\Delta W \propto -\frac{\delta J}{\delta W} = -\frac{\delta J}{\delta T}\frac{\delta T}{\delta W} \approx y(p|e|^{p-1}sign(e))^T$$

(3.27)

Donde *T* es la transpuesta del vector. Por tanto, las operaciones de la red serán: (CURIEL, 2010)

Paso hacia delante:

$$y_i = \sum_{j=1}^{N} W_{ij} X_j, \forall i$$

(3.28)

Paso hacia atrás:

$$e_j = X_j - \sum_{i=1}^{M} W_{ij} y_i$$

(3.29)

Actualización de los pesos:

$$\Delta W_{ij} = ny_i sign(e_j)|e_j|^{p-1}$$

(3.30)

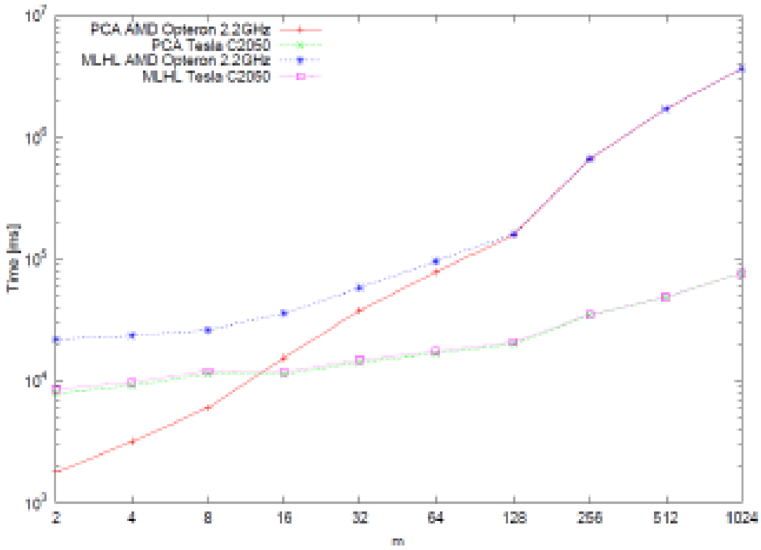


Figura 3.10 Neural PCA (NFN) and MLHL execution time for the 1024-dimensional data set (CORCHADO, 2010)

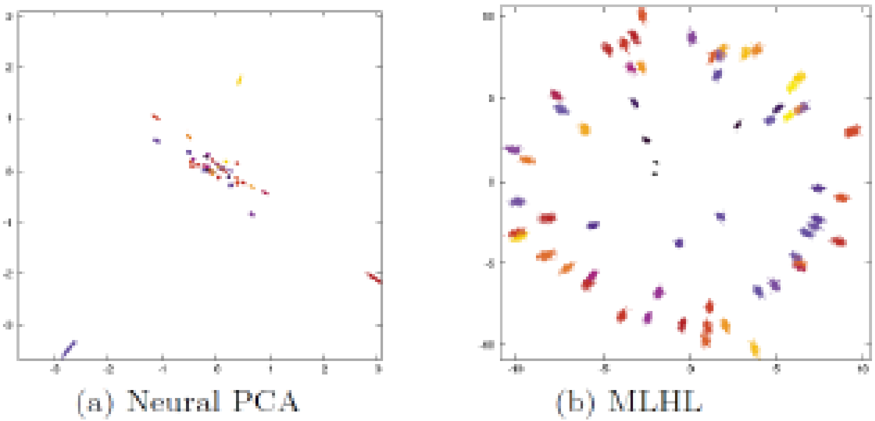


Figura 3.11 The results of projection to 2D for the 1024-dimensional data set (CORCHADO, 2010).

- **APRENDIZAJE COOPERATIVO HEBBIANO PARA LA MAXIMIZACIÓN DE LA PROBABILIDAD DEL RESIDUO - COOPERATIVE MAXIMUN LIKELIHOOD HEBBIAN LEARNING (CMLHL)**

La Distribución Gaussiana Rectificada (DGR) es una modificación de la Distribución Gaussiana estándar **[69]**. Las conexiones laterales se derivan de la DGR y se basan en distribuciones cooperativas. (CORCHADO, 2009; CORCHADO, 2011) Se parte de la DGR estándar:

$$p(y) = Z^{-1}e^{-\beta E(y)} \tag{3.31}$$

[69] http://www.bioestadistica.uma.es/libro/node79.htm (UMA, bioestadística)

Donde el parámetro *"β=1/T"* es la inversa de la temperatura. *"β"* produce un decremento de la temperatura concentrando la distribución al mínimo de la función de energía. El factor *"Z"* normaliza la integral de "p(y)" a la unidad y "E(y)", es la función de coste o energía asociada, expresada como:

$$E(y) = \frac{1}{2}\, y^T\, Ay - b^T y$$

(3.32)

La función cuadrática de energía "E(y)" queda definida por el parámetro "bías" *"b"* y la matriz simétrica "A" . Dependiendo de dicha matriz, "E(y)" podrá tener diferentes curvaturas. En la DGR no todas las funciones de energía se pueden utilizar, sólo si la matriz A cumple la propiedad de la copositividad, mediante la cual se bloqueará las direcciones en la que la energía tiende al infinito negativo. Expresado matemáticamente seria de la siguiente manera:

$$y^T A > 0 \; para \; todo \; y > 0$$

(3.33)

La Distribución Cooperativa en el caso de *N* variables se define como:

$$A_{ij} = \delta_{ij} + \frac{1}{N} - \frac{4}{N}\cos(\frac{2\pi}{N}(i-j))$$

(3.34)

$$b_i = 1$$

(3.35)

Donde "δ_{ij}" es el delta de Kronecker e *"i"* y *"j"* identifican la neurona de salida. Como aproximación, para acelerar el aprendizaje, se puede simplificar la matriz "**A"** como indica la ecuación (3.36) y muestra la Figura 3.12, donde la matriz "**A"** se emplea para modificar la respuesta de los datos basándose en la distancia entre las neuronas. Así mismo la figura en mención nos muestra las 24 salidas, donde los cuadros negros representan los valores negativos y los blancos los valores positivos. (CURIEL, 2010)

$$A_{ij} = (\delta_{ij} - \cos(2\pi(i-j)/N)$$

(3.36)

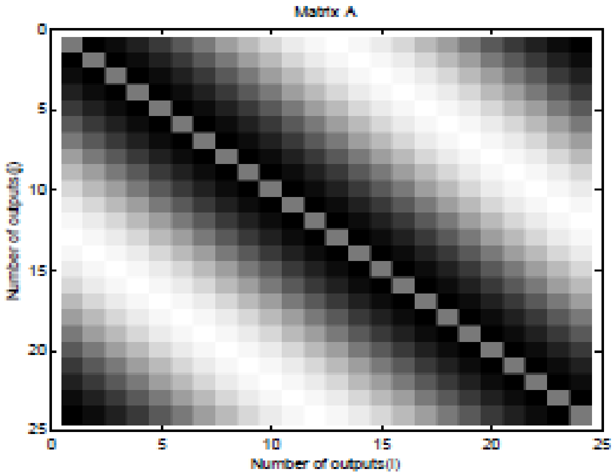


Figura 3.12 Matriz A para la red Gaussiana Rectificada con 24 salidas. Los cuadrados negros representan valores negativos y los blancos valores positivos (CURIEL, 2010).

Las modas de la DGR son los mínimos de la función de energía y caracterizan su comportamiento a baja temperatura. El algoritmo más simple es la proyección del método del gradiente rectificado, que consiste en el gradiente seguido de la siguiente rectificación:

$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+$$

(3.37)

Donde $[\,]^+$ es la rectificación necesaria para que los valores *"y"* permanezcan en el cuadrante positivo y *"t"* la fuerza de las conexiones laterales. Para que el algoritmo converja a un punto estacionario de la función de energía, generalmente un mínimo local, se debe escoger un valor adecuado para *"t"* . La moda de la distribución se puede obtener derivando, aplicando el gradiente descendiente de la función de energía respecto a *"y"*: (CURIEL, 2010)

$$\Delta y \alpha - \frac{\partial E}{\partial y} = -(Ay - b) = b - Ay$$

(3.38)

Se parte del algoritmo para la maximización de la probabilidad del residuo (MLHL) y se añaden conexiones laterales, como se muestra en la Figura 3.13. Estas se derivan de la DGR y actúan en el paso hacia delante y la realimentación.

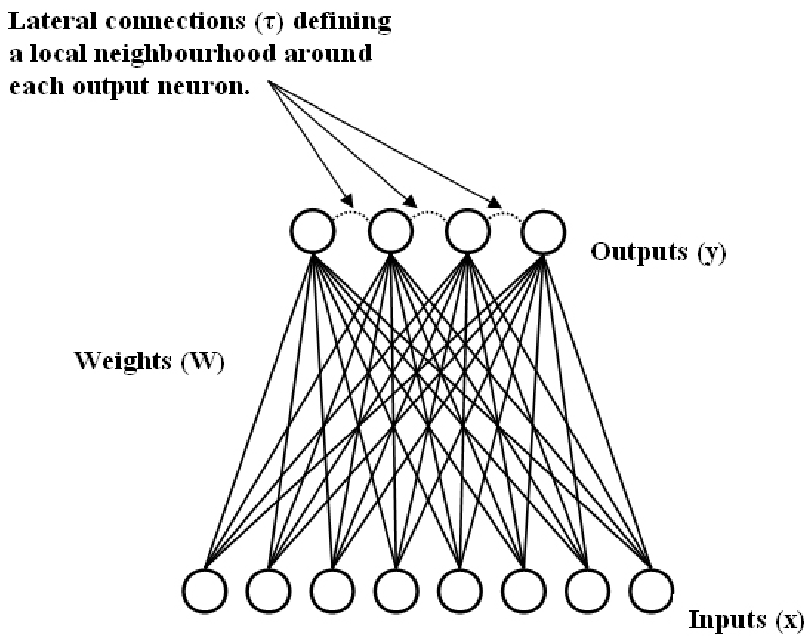


Figura 3.13 CMLHL: lateral connections between neighbouring output neurons (CORCHADO, 2011)

La arquitectura final de la red será capaz de identificar un cierto tipo de orden global o estructura en el conjunto de datos. A esta red se denomina Cooperative Maximum Likelihood Hebbian Learning (CMLHL) y los pasos son:

Paso hacia delante:

$$y_i = \sum_{j=1}^{N} W_{ij} X_j \, , \forall i \quad \textbf{(3.39)}$$

Paso hacia atrás o realimentación:

$$e_j = X_j - \sum_{i=1}^{M} W_{ij} \, y_i \, , \forall j \quad \textbf{(3.40)}$$

Activación lateral:

$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ \quad \textbf{(3.41)}$$

Actualización de los pesos:

$$\Delta W_{ij} = n y_i sign\,(e_j)|e_j|^{p-1} \quad \textbf{(3.42)}$$

3.1.6.2 APRENDIZAJE COMPETITIVO - COMPETITIVE LEARNING

En las redes con aprendizaje competitivo (y Cooperativo), suele decirse que las neuronas compiten (y Cooperan) unas con otras con el fin de llevar a cabo una tarea dada. Con este tipo de aprendizaje se pretende que cuando se presente a la red cierta información de entrada, sólo una de las neuronas de salida de la red, o una por cierto grupo de neuronas, se active (Alcance su Valor de Respuesta Máximo). Por tanto las neuronas compiten para activarse quedando finalmente una, o una por grupo, como neurona vencedora y el resto quedan anulados y siendo forzadas a sus valores de respuesta mínimos.

La competición entre neuronas se realiza en todas las capas de la red, existiendo en estas redes neuronas con conexiones de autoexcitación (Signo Positivo) y conexiones de inhibición (Signo Negativo) por parte de neuronas vecinas.

El objetivo de este aprendizaje es categorizar (Clusterizar) los datos que se introducen en la red, de esta forma las informaciones similares son clasificadas formando parte de la misma categoría y por tanto deben activar la misma neurona de salida. Las clases o categorías deben ser creadas por la propia red, puesto que se trata de un aprendizaje no supervisado a través de las correlaciones entre los datos de entrada

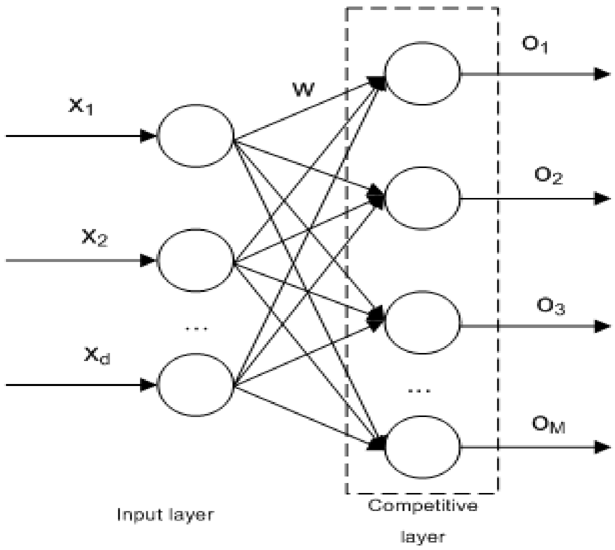


Figura 3.14 Competitive neural network architecture (SALATAS, 2011)

- **MAPAS AUTO-ORGANIZADOS DE KOHONEN – SEL-ORGANIZINF MAP (SOM)**

Es un tipo de red neuronal artificial**[67]**, que es entrenada usando aprendizaje no supervisado**[68]** para producir una representación discreta del espacio de entrada de las muestras entrantes, llamado mapa.

[67] Véase el apartado 3.1
[68] Véase el apartado 3.1.3.3

Los mapas auto-organizados son diferentes de otras redes neurales artificiales, en el sentido que estos usan una función de vecindad para preservar las propiedades topológicas del espacio de entrada.

SOMs son útiles para visualizar vistas de baja dimensión de datos de alta dimensión, semejante a un escalado multidimensional.

El modelo fue descrito por primera vez como una red neural artificial por el profesor finlandés Teuvo Kohonen, debido a lo cual en ocasiones son llamadas redes o mapas de Kohonen.

Como la mayoría de las redes neuronales artificiales, SOMs operan en dos modos: entrenamiento y mapeo. En el entrenamiento construye el mapa usando ejemplos entrantes, mientras el mapeo clasifica una nueva entrada.

La descripción matemática del presente algoritmo de aprendizaje es la siguiente:

Se presenta un vector de entrada "X" y se calcula el nodo ganador:

$$n_i = \arg min_j \{||X - W_j||\} \quad \textbf{(3.43)}$$

Se seleccionan los nodos vecinos de "*nj* (*j*∈*V*(*j**))" y se calcula el valor de la "función de vecindad", "Λ(j*,i)", función decreciente con la distancia en el espacio de salida:

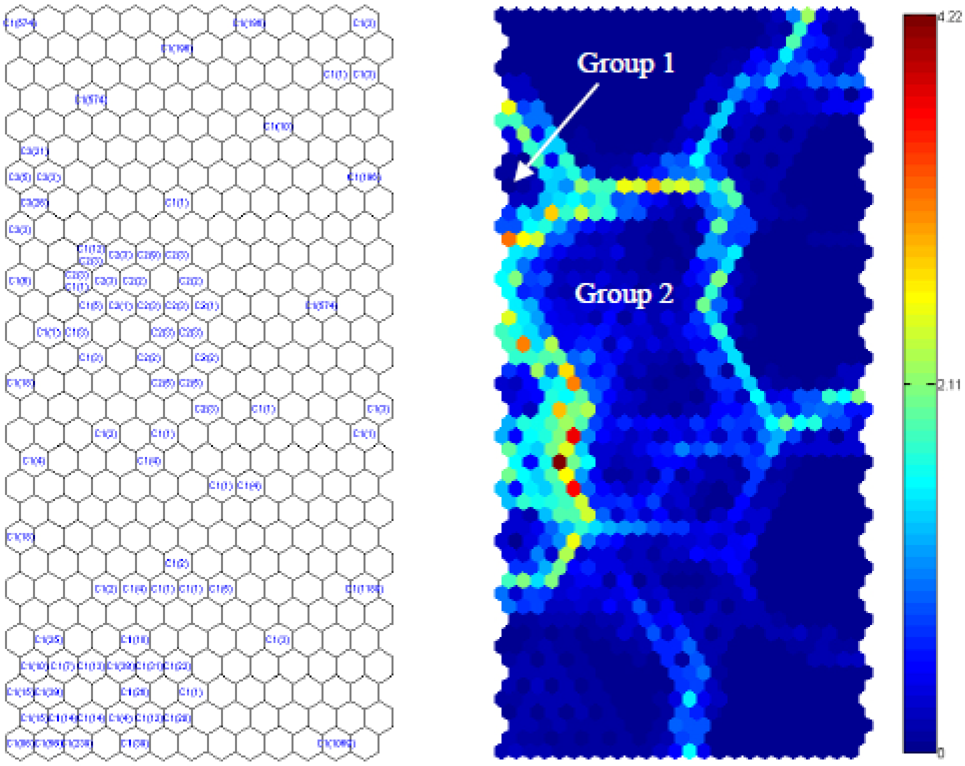
$$\Lambda(j,i) = \exp\left(-\frac{|r_j - r_i|^2}{2\sigma^2}\right) \quad r_j : posición\ de\ n_j ; \sigma : influencia\ de\ la\ vecindad. \quad \textbf{(3.44)}$$

Se actualizan los pesos "w_i" de todas los nodos en "*V*(*j**)"

$$\Delta W_j = n\ \Lambda(j^*,i)(X - W_j) \quad (con\ i \in V\ (j^*)\ "vecindad\ de\ n_{j^*}") \quad \textbf{(3.45)}$$

La convergencia depende de una adecuada elección de η: (p.ej. η= 1/t)

La influencia de la vecindad, "σ" , también debe decrecer con el tiempo. La actualización de los pesos puede ir seguida de una normalización de los pesos "w_i" , pero no siempre es necesario.



a. Labelled map. b. Associated U-matrix.

Figura 3.15 SOM, visualización de las pruebas realizadas en MOVICAB-IDS SYSTEM (CORCHADO, HERRERO, 2013).

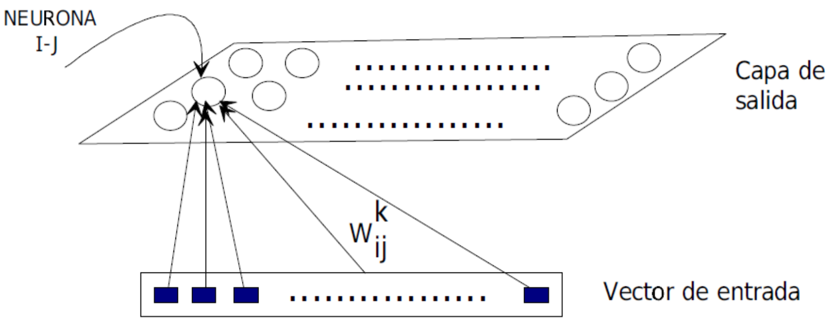


Figura 3.16 Arquitectura típica de un mapa SOM

- **ANALISIS DE COMPONENTE CURVILINEO - CURVILINEAR COMPONENT ANALYSIS (CCA)**

Análisis de componente curvilíneo (CCA), es un método de reducción de dimensionalidad no lineal; fue desarrollado como una mejora sobre los mapas Auto-organizativos (SOM)**[69]**. Esto trata de engañar las limitaciones inherentes en algunos

modelos anteriores lineales como PCA[70]. Su salida no es un enrejado fijo, pero un espacio continuo capaz de tomar la forma del subcolector en el conjunto de datos (espacio de entrada). En la presente memoria no profundizaré respecto al presente algoritmo, pero será considerado para trabajos futuros por lo que recomiendo (CORCHADO, HERRERO, 2013) donde se aborda una explicación más detallada del análisis del componente curvilíneo. CCA es capaz de realizar la reducción de dimensionalidad y representar la estructura intrínseca de datos de entrada, sin cualquier conocimiento anterior sobre la distribución del conjunto de datos analizados. Comparado con otros algoritmos de proyección anteriormente descritos, el método CCA es más general, confiable, y más rápido en la captura de la estructura de datos de entrada.

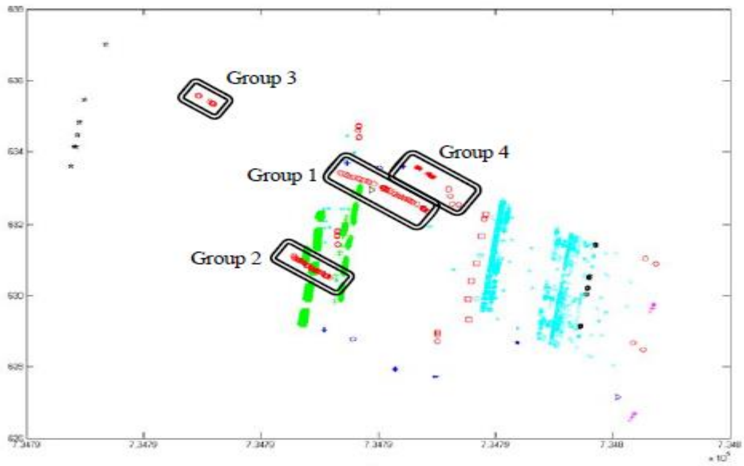


Figura 3.17 CCA, visualización de las pruebas realizadas en MOVICAB-IDS SYSTEM (CORCHADO, HERRERO, 2013).

3.1.7 APRENDIZAJE REFORZADO (REINFORCEMENT LEARNING)

El aprendizaje reforzado, consiste en que este aprenda que “acciones” realizar cuando se encuentra en cierto “estado”, de modo de maximizar una recompensa. El agente interactúa con un entorno definido, y las acciones que este realiza modifican dicho entorno. De aquí se puede desprender dos características de este tipo de aprendizaje: primero, debe definirse el entorno y una representación discreta de este, que corresponda al estado; segundo, el agente debe ser capaz de identificar en qué estado se encuentra y las recompensas asociadas a cada estado. (RICHARD, 1998)

La principal diferencia del aprendizaje reforzado con otros tipos de aprendizaje es que usa como información para el entrenamiento la evaluación de las acciones y el resultado de estas en el entorno, en vez de instruir al agente diciéndole que acción tomar. El aprendizaje reforzado puede ser aplicado a tareas episódicas y a tareas no episódicas. En la Figura 3.18 se observa un esquema básico del funcionamiento del aprendizaje reforzado.

[69] Véase el apartado 3.1.13
[70] Véase el apartado 3.1.6

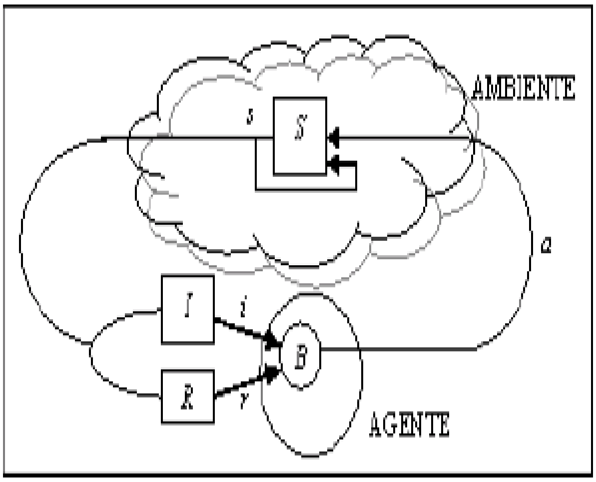


Figura 3.18 Aprendizaje reforzado

3.1.8 VENTAJAS Y LIMITACIONES DE LAS ANN

Las ANN no son la panacea que permite resolver todos los problemas, sino que están orientadas a un determinado tipo de tareas. Podemos destacar cuatro características del problema o tarea que hacen aconsejable la utilización de las ANN (MARTÍN DEL BRÍO Y SANZ, 1997). Por una parte, no se dispone de un conjunto de reglas sistemáticas que describan completamente el problema. En cambio, sí disponemos de muchos ejemplos o casos (condición indispensable para poder aplicar las ANN). Por otra parte, los datos procedentes del problema son imprecisos, incoherentes o con ruido. Por último, el problema es de elevada dimensionalidad, es decir, el número de variables de entrada es demasiado grande como para que un modelo convencional aprenda a solucionar el problema en un tiempo razonable. Cuando no se dan estas circunstancias puede ser más aconsejable optar por solucionar el problema mediante un modelo derivado de la estadística o la Inteligencia Artificial. Por tanto, no debemos concebir las ANN como una alternativa, sino más bien como un complemento a los modelos convencionales ya establecidos. Las ANN presentan una serie de ventajas frente a los modelos estadísticos. Una ventaja fundamental consiste en que los modelos neuronales normalmente no parten de restricciones respecto de los datos de partida (tipo de relación funcional entre variables), ni suele imponer presupuestos (como distribución gaussiana u otras). Por otra parte, la habilidad de las neuronas de calcular funciones de salida no lineales capacita a la red para resolver problemas complejos o no lineales. (PALMER, A ; MONTAÑO, JJ) De este modo, en numerosas aplicaciones se están consiguiendo con ANN cotas de error mucho mejores que las proporcionadas por la estadística. Respecto a las limitaciones que presentan las ANN, una de las más importantes consiste en que es difícil comprender la naturaleza de las representaciones internas generadas por la red para responder ante un problema determinado. Es decir, no sabemos cómo el sistema interrelaciona las diferentes variables de entrada con los pesos de las conexiones entre neuronas para elaborar una solución (RZEMPOLUCK, 1998). Esta limitación contrasta con los diferentes modelos estadísticos, los cuales permiten observar los parámetros o pesos relativos que el modelo otorga a cada una de las variables que intervienen en el modelo. Con el fin de solventar esta limitación y así determinar qué

es lo que la red ha aprendido, algunos autores (por ejemplo, LISBOA, MEHRI-DEHNAVI Y MARTIN, 1994) hacen uso de matrices de sensibilidad, las cuales permiten cuantificar la importancia que tiene cada variable de entrada sobre cada variable de salida de la red.

4. CAPÍTULO CUARTO

"When Julius Caesar sent messages to his generals, he didn't trust his messengers. So he replaced every A in his messages with a D, every B with an E, and so on through the alphabet. Only someone who knew the "shift by 3" rule could decipher his messages… and so we begin."

Phil Zimmerman

Una vez explorada el área de conocimiento de la Detección de intrusiones y de cómo las redes neuronales, en la actualidad cumplen un papel de suma importancia en la eficiencia y efectividad de los IDS; en el presente capítulo propondré una solución para la visualización de comportamientos anómalos en la red basándome en la aplicación de técnicas en Inteligencia Artificial fundamentalmente en las redes neuronales artificiales (ANN). Tal propuesta tiene como base, trabajos de investigación relevantes por la comunidad científica. De este modo se detallarán las etapas de implementación del proyecto, la selección y representación de los datos, la selección del modelo y arquitectura de la red neuronal, la configuración de los parámetros de entrenamiento, la verificación del aprendizaje de la red, las herramientas utilizadas para la simulación de las ANN.

4.1 SOLUCION PROPUESTA PARA LA DETECCION DE INTRUSOS CON REDES NEURONALES

Para que una red neuronal pueda cumplir su propósito, factores como disponibilidad y características de los datos, así como también la topología y el algoritmo de entrenamiento deben ser cuidadosamente analizados (HEADY ,1990). Definido el ámbito del problema que se aborda en el presente proyecto de fin de máster la solución propuesta, se da en ***modelos de inteligencia computacional aplicados al diseño e investigación de sistema de detección de intrusos***. Para su descripción se utiliza el modelo de las redes neuronales artificiales, enunciado anteriormente en el apartado 3.1; específicamente en el aprendizaje no supervisado, apartado 3.1.3.3. Centrándome principalmente en el enfoque de detección de anomalías, estudiando la posibilidad de detectar también nuevos ataques (Zero day attack).

4.2 TRABAJOS RELACIONADOS CON LA PROPUESTA

La aplicación de los modelos de las redes neuronales artificiales (ANN), han demostrado su utilidad en la solución de problemas complejos. Las ANN han sido utilizadas satisfactoriamente en la predicción de diversos problemas en diferentes áreas de conocimiento como son: biología, medicina, economía, ingeniería y psicología (ARBIB, 1995; SIMPSON, 1995; ARBIB, ERDI Y SZENTAGOTHAI, 1997); con buenos resultados respecto a los modelos derivados de la estadística clásica (BONILLA Y PUERTAS, 1997; DUNCAN, 1997;FRENCH, DAWSON Y DOBBS, 1997; JEFFERSON, PENDLETON, LUCAS ET AL., 1997; SHEKHARAN, 1997; TOMMASO, SCIRUICCHIO, BELLOTTI ET AL., 1997; VOHRADSKY, 1997; WEST,

BROCKETT Y GOLDEN, 1997; DE LILLO Y MERAVIGLIA, 1998; JANG, 1998; WALLER, KAISER, ILLIAN ET AL., 1998).

En las soluciones para sistemas de detección de intrusos con base en la detección de anomalías en el tráfico de redes de computadoras son abordadas en diversos trabajos en la literatura especializada. Tenemos por ejemplo el trabajo de Álvaro Herrero y Emilio Corchado en su libro: “Mobile Hybrid Intrusion Detection: The MOVICAB-IDS System” (2013), donde combinan diferentes paradigmas AI; detectando situaciones anómalas logrando la visualización de los paquetes en el tráfico de redes, con un modelo de proyección neuronal en tiempo real. En otro estudio los autores anteriormente mencionados, propusieron idMAS-SQL (2011) una arquitectura basada en la jerarquía y distribución estratégica donde las funcionalidades son estructurados en capas, dicho estudio presenta una arquitectura multiagente que apunta a la detección de ataques de Sql Injection, que es uno de los más frecuentes ataques que se dan en las bases de datos; donde incorpora un nuevo modelo de clasificación basado en una mezcla de red neuronal y las máquinas de soporte vectorial (Support Vector Machine -SVM).

En el trabajo de Mafra y otros (2008) presenta un modelo de sistemas de detección de intrusión que clasifica el tráfico de red por análisis de comportamientos normal o anómalo. Para la detección de anomalías son utilizadas dos técnicas de inteligencia artificial llamadas Support Vector Machine (SVM) y Redes Neuronales Hhonen (KNN).

Xian and Lim (2005) su propuesta se basó en un clasificador hibrido con multiples capas de detecion de intrusión, donde los ataques son clasificados en tres categorías: negación de servicio (DOS), escaneo (PROBE) y otros. La categoría es dividida en dos subcategorías: usuarios locales que intentan obtener privilegios de súper usuario (U2R) y ataques a servicios a través de la red (R2L).

En los trabajos de Zanero (2004), Kayacik (2003) y Lei and Ghorbani (2004) usan la técnica de mapas auto-organizados (SOM) y algunas variaciones para almacenar los datos provenientes del entrenamiento de la red neuronal. La idea principal de abordar las ANN es la disponibilidad de un método de clasificación sin supervisión, rápido y eficiente para una entrada de datos con muchas variables (IP de origen, IP destino, puerto de origen, puerto de destino, tamaño de los paquetes, protocolo, etc). Un problema presente en ANN es el entrenamiento de estas redes, que es normalmente efectuado “off-line”. Con todo, una vez entrenadas, los tiempos de análisis son considerablemente eficientes.

En el trabajo propuesto por Shyu (2003), que utiliza ANN para la detección de intrusión, fue aplicada la técnica del PCA, una técnica de redes neuronales con aprendizaje no supervisado, que busca, que busca la reducción de los datos y facilitar el análisis posterior de los mismos.

4.3 APLICACIÓN DEL MODELO NEURONAL NO SUPERVISADO PARA LA PROYECCCION DEL TRAFICO DE DATOS.

En el presente proyecto de fin de máster trata de brindar un acercamiento diferente a los tradicionales IDS que están basados principalmente desde el punto de vista de clasificador. En esta línea de investigación el objetivo principal es de proveer al administrador de la red una Visualización (snapshot) del tráfico de su red. Mejorando la eficacia de la detección de eventos anómalos para los analistas de la red/seguridad en sus centros operacionales (D'AMICO, 2007). En consecuencia la combinación existente de técnicas de proyección y de "scatter plots"[71] ya constituyen un instrumento de visualización muy útil para investigar la estructura intrínseca del conjunto de datos multidimensionales, permitiendo a expertos poder estudiar las relaciones entre proyecciones diferentes dependiendo de la técnica empleada (CORCHADO,HERRERO, 2013). Esta idea ha sido aplicada en el presente PFM, donde las técnicas de proyección apoyan las búsquedas visuales de estructura (tanto normal como anómalo) en el conjunto de datos obtenidos del tráfico de red.

Existen una gran variedad de herramientas comerciales y de investigación en IDS que brindan la visualización a nivel de paquetes, pero la mayor parte de ellos no proporcionan una visualización intuitiva de paquetes conservando el contexto temporal. Así, requieren de previos entrenamientos para identificar las "huellas digitales visuales" de los ataques. Este problema es vencido en el presente PFM combinando la selección del modelo de proyección neuronal y la técnica del "scatter plots" [71]. Como consecuencia de esta combinación, el contexto temporal de datos (paquetes) es conservado en las vistas y luego proporciona una vista sintética e intuitiva del tráfico de la red. En el caso de la técnica de visualización de coordenadas paralelas[71], como una técnica simple compleja, debió ser desechada en el presente PFM a pesar que puede representar un alto número de dimensiones; lamentablemente, sufre de la oclusión que es debido a los datos que se superponen que pueden dejar tanto segmentos de recta como las etiquetas elegibles. Además, las coordenadas paralelas no pueden mostrar la cantidad de paquetes visualizados. El PCA nos permite obtener un modelo para la reducción del tamaño de los vectores de entrada a la ANN, asegurando que la pérdida de información sea mínima y, en consecuencia, disminuyendo la complejidad del clasificador neuronal y manteniendo estables los tiempos de entrenamiento.

Los resultados experimentales del presente PFM (ver el Capítulo Quinto) superan los resultados antes mencionados, porque las proyecciones CMLHL utilizadas están basadas en la estadística de orden más alta; así el funcionamiento de detección es mejorado y se proporciona una explicación más llena del empleo y los resultados obtenidos con estos métodos de proyección.

---

[71] Véase el apartado 2.4

4.4 ATAQUES DE SNMP (Simple Network Management Protocol)

Dentro de los tres tipos de ataques más comunes detectados por los IDS son: el escaneo de redes, ataques de denegación de servicio y ataques de penetración en el sistema. Estos tipos de ataques pueden ser lanzados localmente o remotamente. En su mayoría las herramientas de seguridad enfocan su atención en ataques externos pero los ataques son tan probables de ocurrir tanto dentro como fuera de nuestra red. Esto es la razón porque los ataques de SNMP deben ser considerados. SNMP es un protocolo de red, transportado habitualmente sobre UDP, que se utiliza para la monitorización de los equipos de red. Guarda la información de gestión en variables almacenadas en bases de datos en los propios equipos que tienen activado este protocolo. Estas variables pueden ser luego consultadas (y en ocasiones configuradas) por las aplicaciones de gestión. En un uso típico de SNMP, uno o más ordenadores llamados administradores tienen la tarea de monitorizar o gestionar a un grupo de hosts o dispositivos en una red. Cada sistema gestionado ejecuta en todo momento un programa llamado *agente* que informa mediante SNMP al administrador. A diferencia de NetFlow, las consultas a los routers mediante este protocolo no afectan a las tareas de encaminamiento debido a que se limita a escribir o leer de bases de datos llamadas MIB, operaciones que requieren un coste computacional muy bajo. En estas bases de datos se pueden encontrar, por ejemplo, la cantidad de datos transmitidos por un enlace (sin discriminar por destino final, es decir, sin almacenar la ruta) a una frecuencia típica de cinco minutos. El problema es que los datos que se obtienen son el valor agregado de todos los flujos que circulan a través del enlace, por lo que es necesario realizar un procedimiento de inferencia para diferenciar los distintos flujos (que son los componentes que sumamos para obtener una matriz de tráfico). El segundo problema proviene del hecho de que los proveedores de servicios de Internet no están dispuestos a publicar estos datos, y es comprensible debido a la competencia entre empresas. Este es uno de los motivos principales en la lentitud de la investigación en este campo. A los investigadores en algunas ocasiones se les permite la captura de las muestras en tiempo real para las propias organizaciones, pero es muy difícil conseguir conjuntos de datos de larga duración.

4.5 HERRAMIENTAS UTILIZADAS PARA LAS SIMULACIONES DE LAS REDES NEURONALES

El uso de una herramienta disponible en el mercado para la simulación de redes neuronales, se justifica por el precio y la complejidad del desarrollo de un software propio para la manipulación de redes neuronales y por la facilidad de obtener herramientas abiertas, que atendiesen a las necesidades del presente proyecto. En este caso la decisión fue el utilizar una licencia de MATLAB R2008b para Windows, que es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M) está disponible para las plataformas: Unix, Windows y Mas OS X. Sin embargo existen otras herramientas de software libre como son: SNNS (Stuttgart Neural Network Simulator), que trata de un simulador desarrollado por la universidad

de Stuttgart, puede ser usado para crear, modelar, entrenar y testear redes neuronales de diversos modelos; y JNNS (Java Neural Network Simulator) basado en el núcleo de SNNS, fue desarrollada por WSI (Instituto de ciencia y computación Wilhelm-Schickard) en Alemania, basado en tecnología java, que le brinda mayor portabilidad y facilidad de uso.

4.6 ETAPAS DE IMPLEMENTACION DEL PFM

El presente PFM está basado en la combinación de diferentes paradigmas de IA (Multiagent system-MAS[72], Case-based reasoning-CBR[73] and Artificial neural networks-ANN[74]) para tratar el flujo de datos continuos del tráfico de la red. Para esto dividimos los masivos datos de tráfico de red en conjuntos de datos limitados y los visualizamos. Así proveeremos al personal de seguridad con un snapshot intuitivo para monitorear los eventos que ocurren en el tráfico de red del sistema. La combinación de los paradigmas antes mencionados permitirá al usuario obtener beneficios de ciertas propiedades como: la generalización, que nos permitirá identificar aquellos ataques nunca antes vistos (ANN), el estudio de experiencias pasadas (CBR) , la reactividad, la proactividad, la sociabilidad y la inteligencia (MAS) que sin duda alguna, facilitará la tarea de ID.



Figura 4.1 Procesos elaborados en el PFM.

Para su implementación, el presente PFM está constituido de 4 etapas, siendo las siguientes:

4.6.1 LA CAPTURA Y SELECCIÓN DE LOS DATOS

Como se observó en los capítulos Primero y Cuarto, del apartado 1.1.3 y el 4.4 respectivamente, es claro que toda amenaza de intrusión inicia con la exploración del sistema objetivo, para de esta forma hacer un recuento de sus posibles vulnerabilidades y así realizar un ataque. La única forma de que el atacante pueda te-

ner conocimiento de éstas vulnerabilidades, es enviando diferentes tipos de tramas utilizadas para tal fin, y analizar la respuesta que se recibe de parte del equipo atacado. El proceso de ID empieza con la captura de los datos que viajan en la red de computadores, por medio de técnicas de sniffing[75]. El tráfico de red proporciona información sobre los datos internamente y entre redes o equipos individuales. Los tipos más comunes de datos son los registros Kerberos[76], TCP/IP, datos de uso de la CPU, registros de eventos, datos de comandos de usuario, datos de navegación en Internet, datos de auditoría del funcionamiento del sistema operativo, registros del IDS / IPS, datos de Netflow[77], y los reportes SNMP. Toda esta información es única y de alto valor para la seguridad de la red, específicamente para la detección y prevención de intrusos. Estos datos nos brindan importante información de referencia sobre el comportamiento de los usuarios y del sistema, los patrones de tráfico, e información de alerta sobre el tráfico anómalo en la red.

El tráfico de la red se compone de paquetes, flujo de red y sesiones. Un paquete es cada uno de los bloques en que se divide, en el nivel de red la información a enviar, es una unidad de datos que se transmite entre un origen y un destino en Internet o en cualquier red TCP/IP. Un flujo de red es una secuencia unidireccional de paquetes entre un origen y un destino variables. Los datos de sesión representan la comunicación entre computadores, entre los host, o entre un computador y un host. Normalmente, una sesión se define por seis elementos:

- Dirección de IP de destino.
- Dirección de IP de origen.
- Tipo de servicio.
- Puerto de destino.
- Puerto de origen.
- El tipo de protocolo.

En el presente PFM nos centraremos en las capa de transporte y de internet, pues en conjunto son la encargadas que el proceso de intercambio de datos entre dos entidades se lleve a cabo de manera efectiva, ya sea porque se encargue del establecimiento de la comunicación e integridad de los datos, o por que determine la ruta entre el origen y el destino o su ubicación. Por tal motivo nos centramos en estas dos capas previamente mencionadas, porque son las que están directamente involucradas en los procesos que aquí se estudian (detección de intrusos). Para profundizar los conceptos que se dan en la presente sección es recomendable leer el RFC[78], a continuación se definirán superficialmente algunos conceptos que serán de utilización en los apartados posteriores:

---

[72] algunos de los componentes son cubiertos como agentes deliberativos capaces de aprender y desarrollar con el entorno
[73] algunos de los agente contenidos en el MAS son conocidos como CBR-BDI (CARRASCOSA, 2008) agentes, porque ellos integran el BDI (BRATMAN, 1987) modelo y el paradigma CBR
[74] Véase el apartado 3.1
[75] Un sniffer es un programa para monitorear y analizar el tráfico en una red de computadoras, detectando los cuellos de botellas y problemas que existan en ella.
[76] Kerberos es un protocolo de autenticación de red. Como protocolo de seguridad, usa criptografía de claves simétricas, esto significa que la clave utilizada para cifrar es la misma clave utilizada para descifrar o autenticar usuarios. Esto permite a dos computadores en una red insegura, demostrar su identidad mutuamente de manera segura.
[77] Netflow es un protocolo desarrollado por CISCO Systems para coleccionar información del tráfico de redes.
[78] RFC o Request for Comments ("Petición De Comentarios") son una serie de documentos que empezaron a publicarse en 1969 donde se especifica todo lo relacionado con las tecnologías que soportan la Internet, tales como protocolos, recomendaciones, comunicaciones, interfaces, etc.

- TTL (Time to Live - 8 bits): Especifica el tiempo máximo en segundos que un paquete puede estar circulando por la red. Cada vez que algún nodo recibe y procesa este paquete disminuye su valor como mínimo en 1 segundo. Cuando llega a 0, el paquete no será reenviado.

- TOS (Type Of Service - 8 bits): Nos especifica los parámetros sobre la calidad de servicio que debe proporcionársele a la trama al ser transmitida por la red. Algunas redes pueden darle prioridad a servicios, ya que según la información contenida en este campo clasifica algunos paquetes como más importantes que otros (algunas redes admiten sólo los paquetes con una prioridad cuando esta se encuentra en sobrecarga).

  Estos 8 bits se agrupan de la siguiente manera:

  Bits 0-2: Prioridad: Valores altos para prioridades superiores.
  Bit 3: 0 = Retraso Normal, 1 = Bajo Retraso.
  Bit 4: 0 = Tránsito Normal, 1 = Tránsito Rápido.
  Bit 5: 0 = Fiabilidad Normal, 1 = Alta Fiabilidad.
  Bits 6-7: Reservados para futuros usos.

- Source IP address (32 bits): Este campo representa la dirección IP del equipo remitente y permite que el destinatario responda.

- Destination IP address (32 bits): Dirección IP del destino del mensaje.

- Source port (16 bits): Aquí se especifica el puerto que está usando la aplicación que solicita la transferencia en la maquina origen. El número de puerto de la aplicación, tiene solamente significado local ya que el sistema asigna un número cada vez que una aplicación en el origen solicita el envío de datos. Normalmente van en el rango entre 1024 y 5000.

- Destination port (16 bits): Aquí se especifica el puerto que está usando la aplicación que recibe la transferencia de la maquina origen. En el destino estos puertos son fijos, y se denomina número de puerto bien conocido (well-known port numbers). Normalmente van en el rango entre 1 y 1023.

- Acknowledgment information (control bit and number): en el caso de TCP, la recepción de paquetes es reconocida atrás al remitente.

- Size: Este campo indica el tamaño total en bytes del datagrama, incluyendo el tamaño de la cabecera y el de los datos. El tamaño máximo de los datagramas es normalmente de 576 bytes (64 de cabeceras y 512 de datos). Una máquina en general no envía datagramas de mayor tamaño a no ser que tenga la seguridad de que estos van a ser aceptados por la máquina destino. En caso de que exista fragmentación este campo nos indica el tamaño del fragmento, no el del datagrama original.

- Timestamp: es una secuencia de caracteres, que denotan la hora y fecha (o alguna de ellas) en la cual ocurrió determinado evento.

Una vez estudiadas las diversas alternativas para la captura de datos, sobre la cual se pudiera trabajar y que permita entrenar al algoritmo de aprendizaje. Para esta tarea se utilizó el programa de distribución gratuita Wireshark[79] de donde una vez realizadas las capturas, el mencionado programa posee la ventaja de exportar los datos en formato CSV de donde se direccionaban a una base de datos para su tratamiento posterior. Adicionalmente pueden ser consideradas otras alternativas para la captura de los datos en la red como son Netflow[80] y Netmon[81] para redes inalámbricas; en este caso los datos serian pre-procesados en un diferente camino.

En el presente PFM los datos utilizados para el entrenamiento y testeo de la ANN, corresponden a la base de datos generadas por el autor a través de la captura, durante 6 semanas, de todos los paquetes TCP/IP de una red real. No obstante cabe resaltar que existe también un conjunto de datos que han sido generados para el concurso de la KDDCup(1999)[82] que está a disposición para su utilización en la web. Así mismo desde el sitio web del Laboratorio del MIT se permite la descarga gratuita de estos archivos con la estructura tcpdump correspondiente a ejemplos de ataques y trafico inofensivos de 1998 y 1999.

En la figura 4.2 podemos observar la captura de los datos del tráfico en la red, en el preciso instante en la que se está sufriendo un ataque de denegación de servicio (DOS).

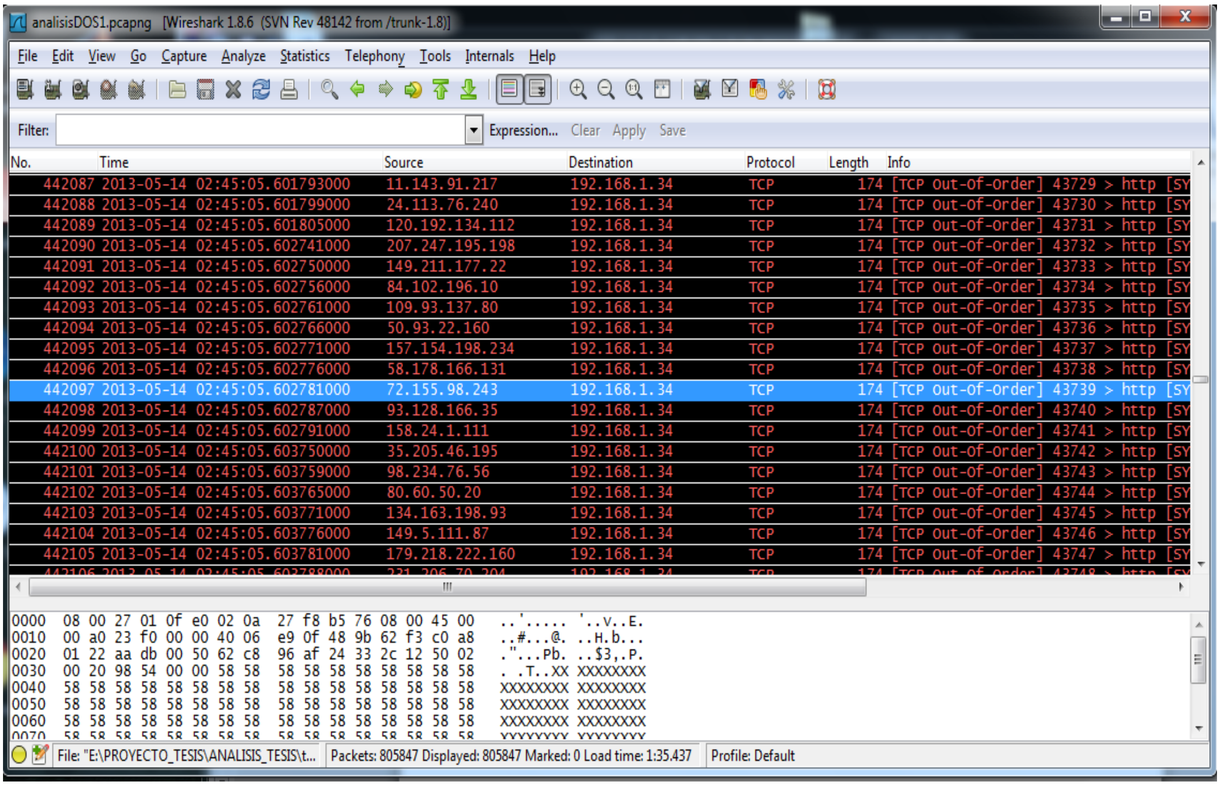


Figura 4.2 Captura del tráfico de datos de la red con Wireshark para su utilización en el PFM.

Posteriormente a la captura se procede a la selección y exportación de los datos, como se mencionó anteriormente para la comodidad del autor se tomó la de-

cisión de tratar los datos en formato CSV. En la figura 4.3 y 4.4 se dan muestra de ello:

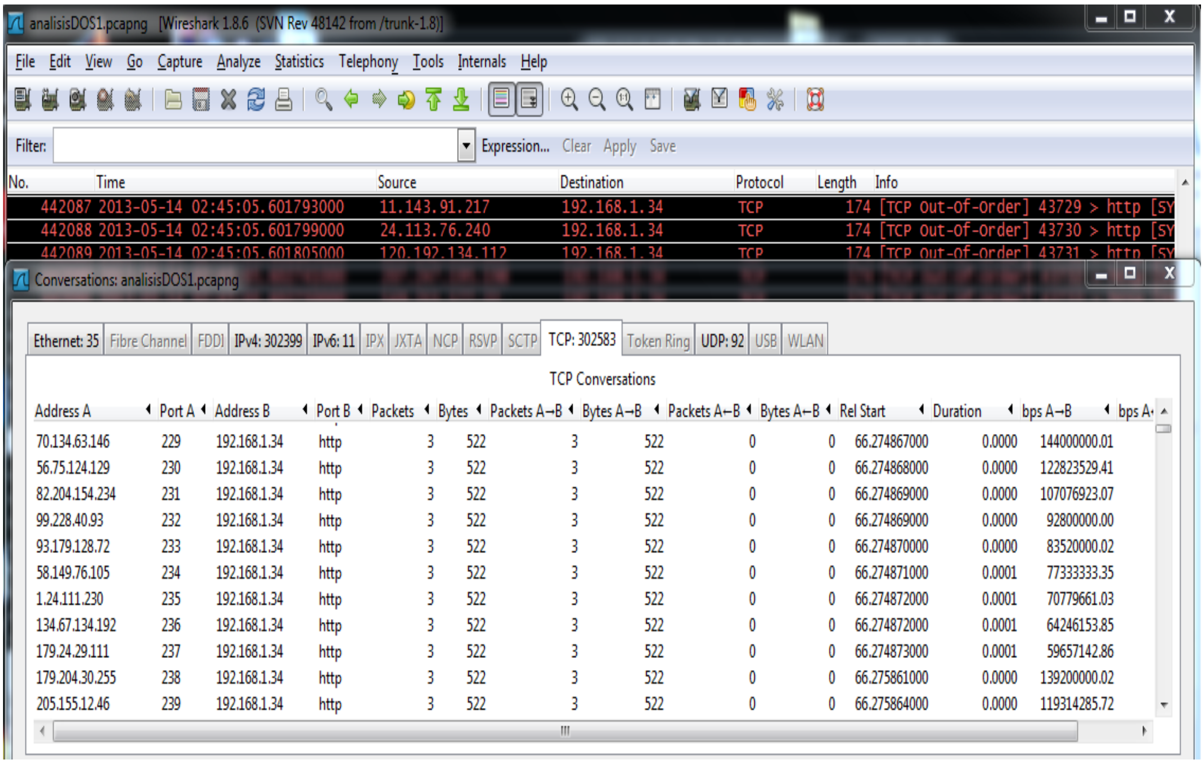
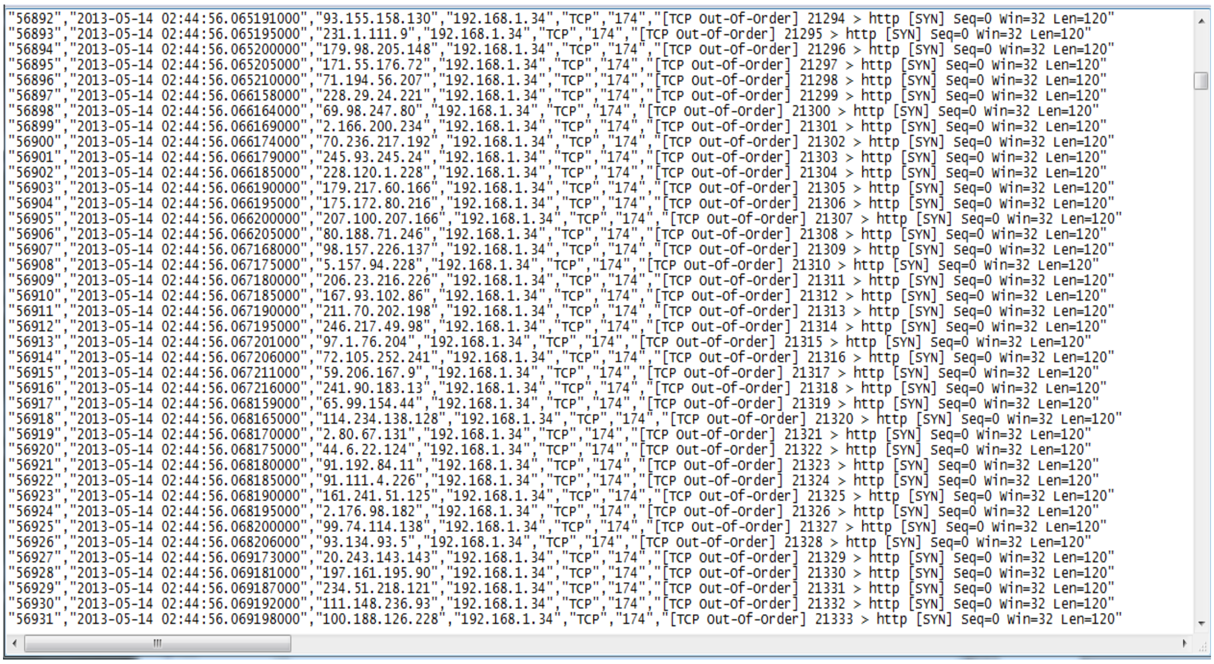


Figura 4.3 Selección de los datos del tráfico de la red para su utilización en el PFM.



```
"56892","2013-05-14 02:44:56.065191000","93.155.158.130","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21294 > http [SYN] Seq=0 Win=32 Len=120"
"56893","2013-05-14 02:44:56.065195000","231.1.111.9","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21295 > http [SYN] Seq=0 Win=32 Len=120"
"56894","2013-05-14 02:44:56.065200000","179.98.205.148","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21296 > http [SYN] Seq=0 Win=32 Len=120"
"56895","2013-05-14 02:44:56.065205000","171.55.176.72","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21297 > http [SYN] Seq=0 Win=32 Len=120"
"56896","2013-05-14 02:44:56.065210000","71.194.56.207","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21298 > http [SYN] Seq=0 Win=32 Len=120"
"56897","2013-05-14 02:44:56.066158000","228.29.24.221","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21299 > http [SYN] Seq=0 Win=32 Len=120"
"56898","2013-05-14 02:44:56.066164000","69.98.247.80","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21300 > http [SYN] Seq=0 Win=32 Len=120"
"56899","2013-05-14 02:44:56.066169000","2.166.200.234","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21301 > http [SYN] Seq=0 Win=32 Len=120"
"56900","2013-05-14 02:44:56.066174000","70.236.217.192","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21302 > http [SYN] Seq=0 Win=32 Len=120"
"56901","2013-05-14 02:44:56.066179000","245.93.245.24","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21303 > http [SYN] Seq=0 Win=32 Len=120"
"56902","2013-05-14 02:44:56.066185000","228.120.1.228","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21304 > http [SYN] Seq=0 Win=32 Len=120"
"56903","2013-05-14 02:44:56.066190000","179.217.60.166","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21305 > http [SYN] Seq=0 Win=32 Len=120"
"56904","2013-05-14 02:44:56.066195000","175.172.80.216","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21306 > http [SYN] Seq=0 Win=32 Len=120"
"56905","2013-05-14 02:44:56.066200000","207.100.207.166","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21307 > http [SYN] Seq=0 Win=32 Len=120"
"56906","2013-05-14 02:44:56.066205000","80.188.71.246","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21308 > http [SYN] Seq=0 Win=32 Len=120"
"56907","2013-05-14 02:44:56.067168000","98.157.226.137","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21309 > http [SYN] Seq=0 Win=32 Len=120"
"56908","2013-05-14 02:44:56.067175000","5.157.94.228","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21310 > http [SYN] Seq=0 Win=32 Len=120"
"56909","2013-05-14 02:44:56.067180000","206.23.216.226","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21311 > http [SYN] Seq=0 Win=32 Len=120"
"56910","2013-05-14 02:44:56.067185000","167.93.102.86","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21312 > http [SYN] Seq=0 Win=32 Len=120"
"56911","2013-05-14 02:44:56.067190000","211.70.202.198","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21313 > http [SYN] Seq=0 Win=32 Len=120"
"56912","2013-05-14 02:44:56.067195000","246.217.49.98","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21314 > http [SYN] Seq=0 Win=32 Len=120"
"56913","2013-05-14 02:44:56.067201000","97.1.76.204","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21315 > http [SYN] Seq=0 Win=32 Len=120"
"56914","2013-05-14 02:44:56.067206000","72.105.252.241","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21316 > http [SYN] Seq=0 Win=32 Len=120"
"56915","2013-05-14 02:44:56.067211000","59.206.167.9","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21317 > http [SYN] Seq=0 Win=32 Len=120"
"56916","2013-05-14 02:44:56.067216000","241.90.183.13","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21318 > http [SYN] Seq=0 Win=32 Len=120"
"56917","2013-05-14 02:44:56.068159000","65.99.154.44","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21319 > http [SYN] Seq=0 Win=32 Len=120"
"56918","2013-05-14 02:44:56.068165000","114.234.138.128","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21320 > http [SYN] Seq=0 Win=32 Len=120"
"56919","2013-05-14 02:44:56.068170000","2.80.67.131","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21321 > http [SYN] Seq=0 Win=32 Len=120"
"56920","2013-05-14 02:44:56.068175000","44.6.22.124","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21322 > http [SYN] Seq=0 Win=32 Len=120"
"56921","2013-05-14 02:44:56.068180000","91.192.84.11","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21323 > http [SYN] Seq=0 Win=32 Len=120"
"56922","2013-05-14 02:44:56.068185000","91.111.4.226","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21324 > http [SYN] Seq=0 Win=32 Len=120"
"56923","2013-05-14 02:44:56.068190000","161.241.51.125","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21325 > http [SYN] Seq=0 Win=32 Len=120"
"56924","2013-05-14 02:44:56.068195000","2.176.98.182","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21326 > http [SYN] Seq=0 Win=32 Len=120"
"56925","2013-05-14 02:44:56.068200000","99.74.114.138","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21327 > http [SYN] Seq=0 Win=32 Len=120"
"56926","2013-05-14 02:44:56.068206000","93.134.93.5","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21328 > http [SYN] Seq=0 Win=32 Len=120"
"56927","2013-05-14 02:44:56.069173000","20.243.143.143","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21329 > http [SYN] Seq=0 Win=32 Len=120"
"56928","2013-05-14 02:44:56.069181000","197.161.195.90","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21330 > http [SYN] Seq=0 Win=32 Len=120"
"56929","2013-05-14 02:44:56.069187000","234.51.218.121","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21331 > http [SYN] Seq=0 Win=32 Len=120"
"56930","2013-05-14 02:44:56.069192000","111.148.236.93","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21332 > http [SYN] Seq=0 Win=32 Len=120"
"56931","2013-05-14 02:44:56.069198000","100.188.126.228","192.168.1.34","TCP","174","[TCP Out-Of-Order] 21333 > http [SYN] Seq=0 Win=32 Len=120"
```

Figura 4.4 Exportación de los datos del tráfico de la red en formato CSV para su utilización en el PFM.

4.6.2 SEGMENTACION

Para la presente etapa, se ha considerado el modelamiento que se da en (CORCHADO, HERRERO, 2013), donde debido a lo complejo y extenso que resultan los datos capturados del tráfico de la red para su tratamiento. Se propone la segmentación de dos formas para su pre-procesado, definiéndolas de la siguiente manera:

- Segmentos iguales simples (Equal Simple Segments - S_X): Tal división del segmento contiene todos los paquetes por fechas, limitado por un tiempo inicial y final. Debe haber una superposición (Overlaps) de tiempo entre cada par de segmentos consecutivos simples porque evidentemente situaciones anómalas podrían ocurrir entre el segmento simple Sx y Sx+1.

- Segmentos acumulados (Accumulated segments - A_X): Cada uno de estos contiene varios segmentos consecutivos simples. Para evitar paquetes duplicados, la superposición (overlaps) de tiempo es removido.

En la Figura 4.5 vemos la representación gráfica de las dos formas de segmentación anteriormente definidas. Como se comentó anteriormente los datos del tráfico de red son divididos en diversos segmentos, en la presente etapa experimental se tomó la decisión de que los segmentos iguales simples (S_X) tendrían una duración de diez (10') minutos y la superposición (Overlaps) entre dos segmentos simples consecutivos será la de dos (02') minutos. En la Tabla 4.1 podemos visualizar como los datos han sido segmentados, para que luego dichos segmentos sean tratados en la etapa del pre-procesado.

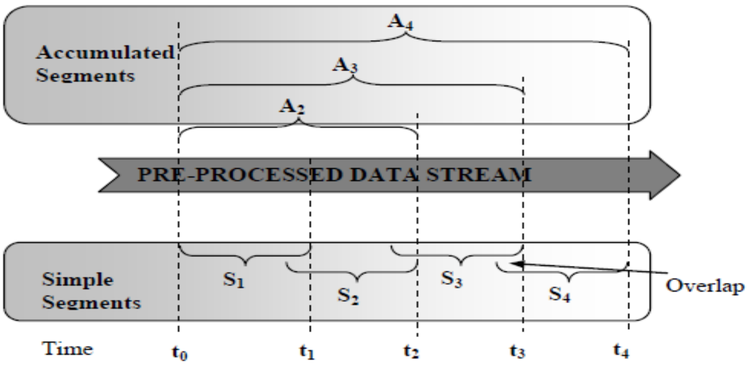


Figura 4.5 Segmentación de los datos para el pre-procesado (CORCHADO, HERRERO, 2013).

[79] Wireshark, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica para educación. Cuenta con todas las características estándar de un analizador de protocolos.
[80] NetFlow, es un protocolo de red desarrollado por Cisco Systems para recolectar información sobre tráfico IP. Netflow se ha convertido en un estándar de la industria para monitorización de tráfico de red, y actualmente se está soportado para varias plataformas además de Cisco IOS y NXOS, como por ejemplo en dispositivos de fabricantes como Juniper, Enterasys Switches, y en sistemas operativos como Linux, FreeBSD, NetBSD y OpenBSD.
[81] Netmon, es un software que permite realizar capturas de tramas en redes inalámbricas 802.11a/b/g sobre Windows 7.
[82] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[83] Los ficheros CSV (del inglés *comma-separated values*) son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas (o punto y coma en donde la coma es el separador decimal: España, Francia, Italia...) y las filas por saltos de línea. Los campos que contengan una coma, un salto de línea o una comilla doble deben ser encerrados entre comillas dobles.

Dataset	Number of Packets	Time		Length
		Inicial	Final	
S1	7757	2013-05-20 01:15:09.313303000	2013-05-20 01:25:08.610855000	1673188
S2	36554	2013-05-20 01:23:08.535834000	2013-05-20 01:33:08.746639000	3439744
S3	22236	2013-05-20 01:31:08.066355000	2013-05-20 01:41:08.670611000	2626640
S4	8777	2013-05-20 01:39:08.078321000	2013-05-20 01:49:08.726628000	1855891
S5	8178	2013-05-20 01:47:08.435906000	2013-05-20 01:57:08.782779000	1704832
S6	12359	2013-05-20 01:55:08.323769000	2013-05-20 02:05:08.838870000	2076338
S7	7937	2013-05-20 02:03:08.148967000	2013-05-20 02:13:08.894970000	1617166
S8	106552	2013-05-20 02:11:08.880965000	2013-05-20 02:21:08.951103000	780402
S9	149209	2013-05-20 02:19:08.081363000	2013-05-20 02:27:17.827459000	26151464
…	…	…	…	…
A2	42811	2013-05-20 01:15:09.313303000	2013-05-20 01:33:08.746639000	4782733
A3	49358	2013-05-20 01:15:09.313303000	2013-05-20 01:41:08.670611000	6281079
A4	56263	2013-05-20 01:15:09.313303000	2013-05-20 01:49:08.726628000	7642222
A5	62951	2013-05-20 01:15:09.313303000	2013-05-20 01:57:08.782779000	9015592
A6	73003	2013-05-20 01:15:09.313303000	2013-05-20 02:05:08.838870000	10592247
A7	79044	2013-05-20 01:15:09.313303000	2013-05-20 02:13:08.894970000	11857521
A8	184097	2013-05-20 01:15:09.313303000	2013-05-20 02:21:08.951103000	19369339
A9	326736	2013-05-20 01:15:09.313303000	2013-05-20 02:27:17.827459000	44735121
…	…	…	…	…

Tabla 4.1 Segmentación de los datos para el pre-procesado para su utilización en el PFM.

Como hemos podido apreciar en la tabla 4.1 los segmentos S_x y Ax llevan una estrecha relación definidas por el número de paquetes, el tiempo y el tamaño de cada segmento de los paquetes. En las Figuras 4.6 y 4.7 se aprecia claramente lo dicho.

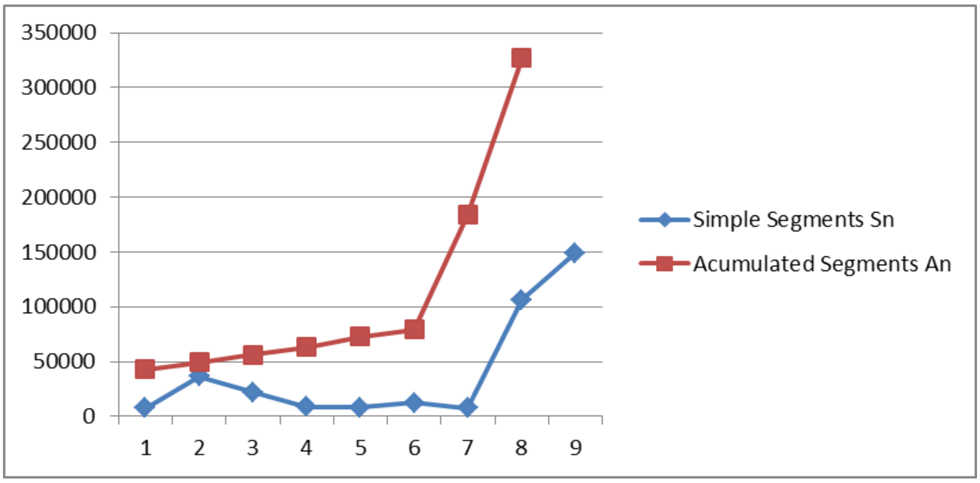


Figura 4.6 Segmentación de Pre-Procesado de Datos (Nº Paquetes).

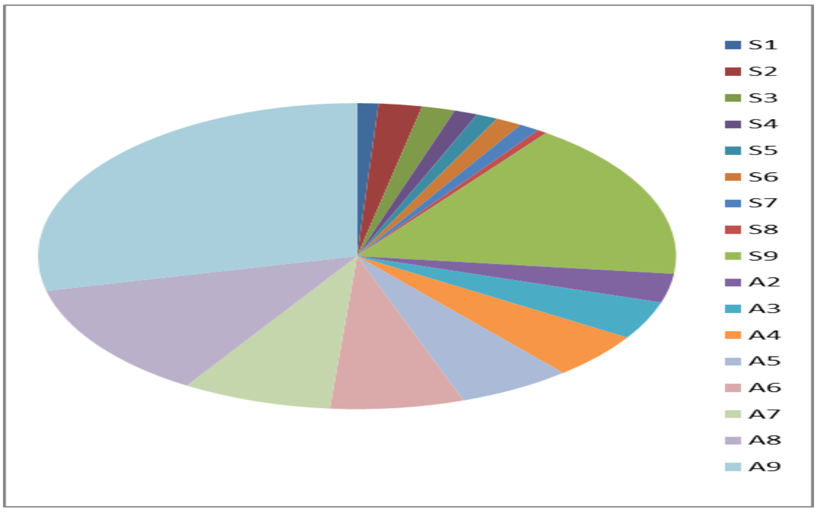


Figura 4.7 Segmentación de Pre-Procesado de Datos (Length).

Tomando en cuenta que los ataques (ANEXO F) han sido realizados localmente, por la brevedad, sólo algunos de estos segmentos fueron seleccionados para la inclusión de sus visualizaciones en el presente PFM. El tráfico contenido en los segmentos seleccionados a continuación puede ser descrito de la siguiente manera:

- S1: Contiene el tráfico normal de la red, por tal motivo en este segmento no encontraremos situaciones anómalas.

- S2: Además del tráfico normal, este segmento contiene un escaneo del tipo de red. Siendo la IP del atacante el 192.168.1.36.

- S3: Además del tráfico normal, este segmento contiene un escaneo de red agregando señuelos al proceso. La IP de los dos señuelos utilizados son: 192.168.1.37 y 192.168.1.39, la IP del atacante y de la víctima son los mismos que la del anterior segmento.

- S4 y S5: Además del tráfico normal, este segmento contiene un ataque del tipo brute force haciendo uso del programa Hydra, la IP del atacante y de la víctima son los mismos que la del anterior segmento.

- S6 y S7: Además del tráfico normal, este segmento contiene un ataque de penetración en el sistema, al servicio ssh puerto 22, donde el atacante con IP 192.168.1.36 obtiene privilegios de Super-usuario accediendo remotamente y haciendo ping a la IP 173.194.41.24 (Google).

- S8: Además del tráfico normal, este segmento contiene una ataque del tipo DOS-flood donde se hace la inundación de paquetes TCP-SYN flags a la IP víctima 192.168.1.40.

- S9: Además del trafico normal, este segmento contiene un ataque del tipo

DOS-flood donde se hace la inundación de paquetes UDP flags a la IP víctima 192.168.1.40

- S10: Además del tráfico normal, este segmento contiene el ataque de una Botnet (localmente), donde la IP de la víctima es el 192.168.1.39 y la IP del atacante el 192.168.1.46

4.6.3 PRE-PROCESADO Y ANALISIS DE LOS DATOS

Después de haber realizado las segmentaciones de los datos del tráfico de la red, los segmentos generados deben ser pre-procesados para aplicar el análisis subsecuente. En esta etapa consistió en la selección de ciertos datos como son: IP Source, IP Destination, Protocols, Length and Number of Packets; a continuación se procedió a diseñar en código un Parser (ver ANEXO E) en vista que dichas variables eran alfanuméricas y para poder procesarlo en Matlab (el cual admite sólo valor numérico); se le asigna por medio de dicho programa un determinado valor. Vemos así que los datos suministrados en la etapa de la segmentación son más ligeros para su presentación y tratamiento posterior, a los algoritmos de las ANN en Matlab. En la Figura 4.8, 4.9 y 4.10 podemos apreciar el preciso instante en la que se pre-procesan los datos haciendo usa de la herramienta de software Matlab.

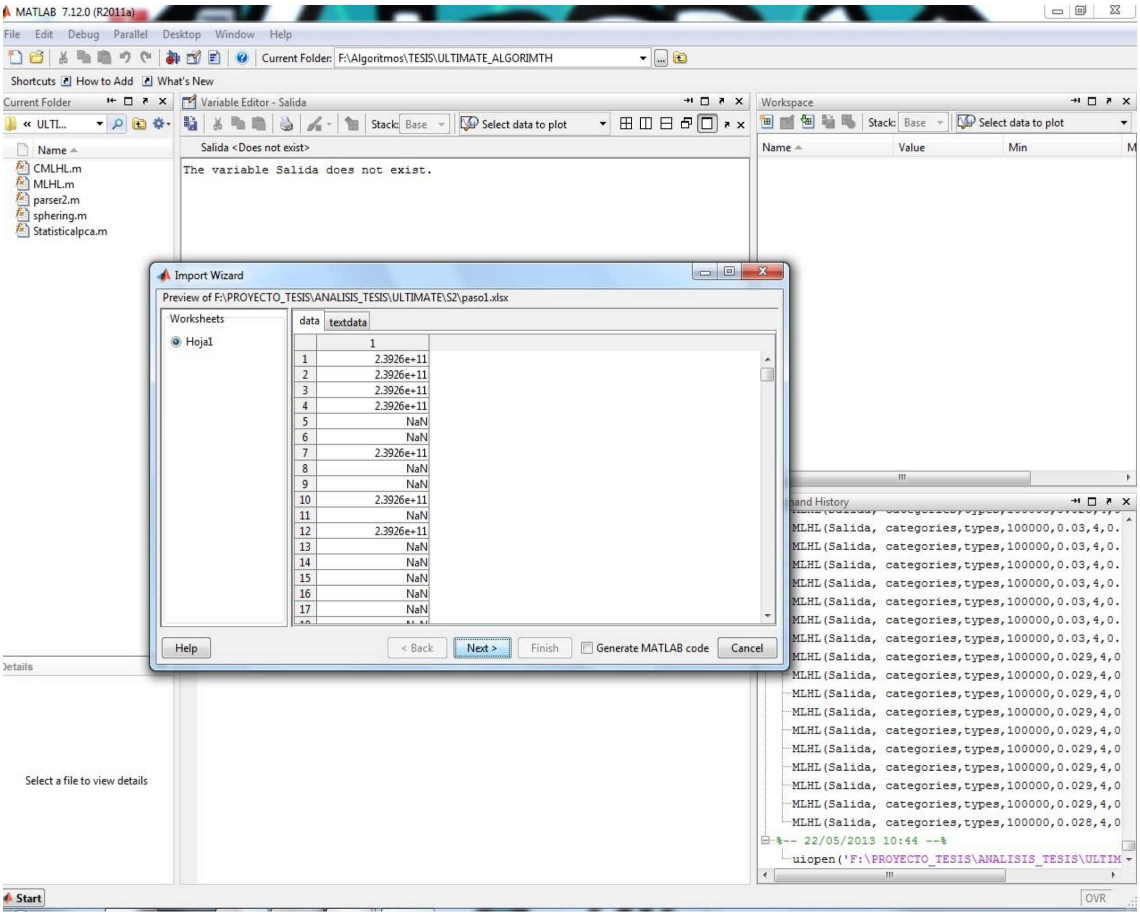


Figura 4.8 Etapa de Pre-procesado en Matlab.

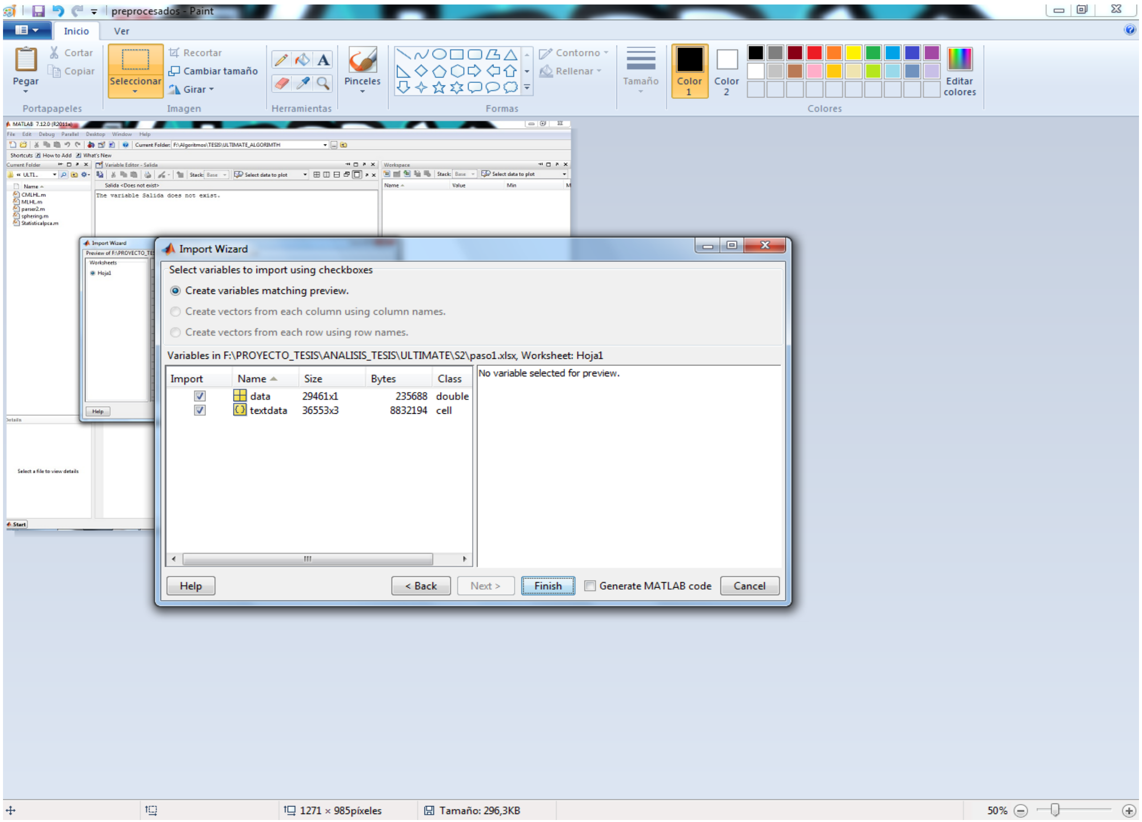


Figura 4.9 Etapa de Pre-procesado en Matlab (Importando datos)

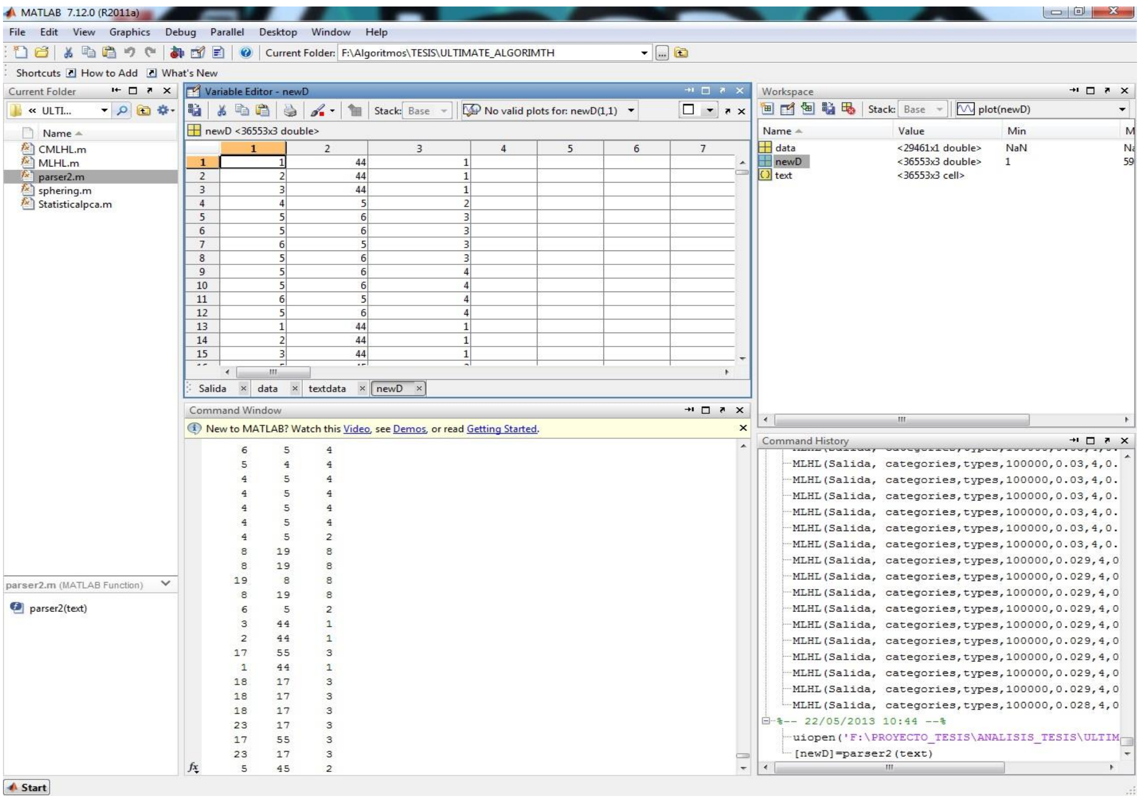


Figura 4.10 Etapa de Pre-procesado en Matlab (Parseado de los datos)

.

En el análisis nos apoyaremos del estudio de caso de experiencia (CBR) para afinar el algoritmo CMLHL (ver apartado 3.1.14) para el análisis de los datos pre-procesados. Se generará una solución (o alcanza estos objetivos) recuperando un caso analizado anteriormente y analizando el nuevo por la arquitectura del CMLHL. Los casos son definidos por varios rasgos (CORCHADO, HERRERO, 2013), como lo muestra la Tabla 4.2:

Class	Feature	Type	Description
P	Segment length	Integer	Total segment length (in ms).
P	Network segment	Integer	Network segment where the traffic comes from.
P	Date	Date	Date of capturing.
P	#source ports	Integer	Total number of source ports.
P	#destination ports	Integer	Total number of destination ports.
P	#protocols	Integer	Total number of protocols.
P	#packets	Integer	Total number of packets.
P	Protocol/packets	Array	An array (of variable length depending on each dataset) containing information about how many packets of each protocol there are in the dataset.
S	#iterations	Integer	Number of iterations.
S	Learning rate	Float	Learning rate.
S	p	Float	CMLHL parameter.
S	Lateral strength	Float	CMLHL parameter.
S	Weights	Matrix	A matrix containing the synaptic weights calculated by the CMLHL model after training.

Tabla 4.2 Representación de características. Clases: P (atributo de descripción problema) y S (atributo de descripción de solución) (CORCHADO, HERRERO, 2013).

El análisis incorpora dos comportamientos diferentes que son: "aprendiendo" "y la explotación". Al principio, durante la etapa de estructuración, se incorpora el nuevo conocimiento (modelado como los juegos de problema/solución) en el caso base como fue descrito previamente. Una vez que la base de datos (Base case) es bastante amplia, el comportamiento de explotación es iniciado. De aquí en adelante, el revisar (Revise) y el retener (Retain), del ciclo CBR no son más realizados. Cuando una nueva petición de análisis llega, el analizador recupera (Retrieve) el caso más similar almacenado previamente en la base de datos (Base case). Entonces los pesos, contenidos en la solución son reutilizados (Reuse) para proyectar los nuevos datos. Otros parámetros del modelo neuronal no son reutilizados, como también la red neuronal no es entrenada otra vez. (CORCHADO, HERRERO, 2013).

En la Figura 4.11 podemos apreciar la representación de las técnicas e instrumentos usados por el analizador, para poner en práctica las etapas del CBR (Recuperación, reutilización, revisión y la retención) que son definidas de la siguiente manera:

- Recuperar (Retrieve): Efectuada la solicitud para el análisis, el analizador tratará de localizar el caso más similar al nuevo en la base de datos (Case base). La recuperación asociativa (MARKS, 2004) basado en la distancia de Euclides es usada para encontrar el caso más similar en el espacio multidimensional definido por los principales rasgos que caracterizan cada caso (ver Tabla 4.2).

- Reutilización (Reuse): una vez que el caso más similar ha sido seleccionado, su solución es reutilizada. Esta solución consiste en los valores de los parámetros usados para entrenar el modelo de CMLHL (ver Tabla 4.2). Un juego de entrenamientos (para el modelo de CMLHL con una combinación de diferentes parámetros de valores que varían en un rango especificado) es propuesto considerando la distancia entre el nuevo caso y el más similar. En otras palabras, si ellos son muy similares, un juego reducido de entrenamientos será realizado. Al contrario, si el caso más similar es lejos del nuevo, un número más grande de entrenamientos con muy diferentes valores de los parámetros será generado.

- Revisar (Revise): El modelo CMLHL es entrenado por el nuevo segmento de tráfico usando la combinación de valores de los parámetros generados en la etapa de reutilización. Cuando la nueva proyección (las salidas del modelo de CMLHL para cada combinación del parámetro) del segmento están listas, son mostradas al usuario. Entonces el usuario selecciona una de estas proyecciones como la mejor; el que proporciona la foto (snapshot) más clara de la evolución del tráfico.

- Retener (Retain): Una vez que la mejor proyección ha sido seleccionada por el usuario, el analizador almacena el nuevo caso conteniendo la descripción del problema y la solución (valores de parámetro usados para generar la proyección escogida) en la base de datos para su futura reutilización (ver Tabla 4.2).

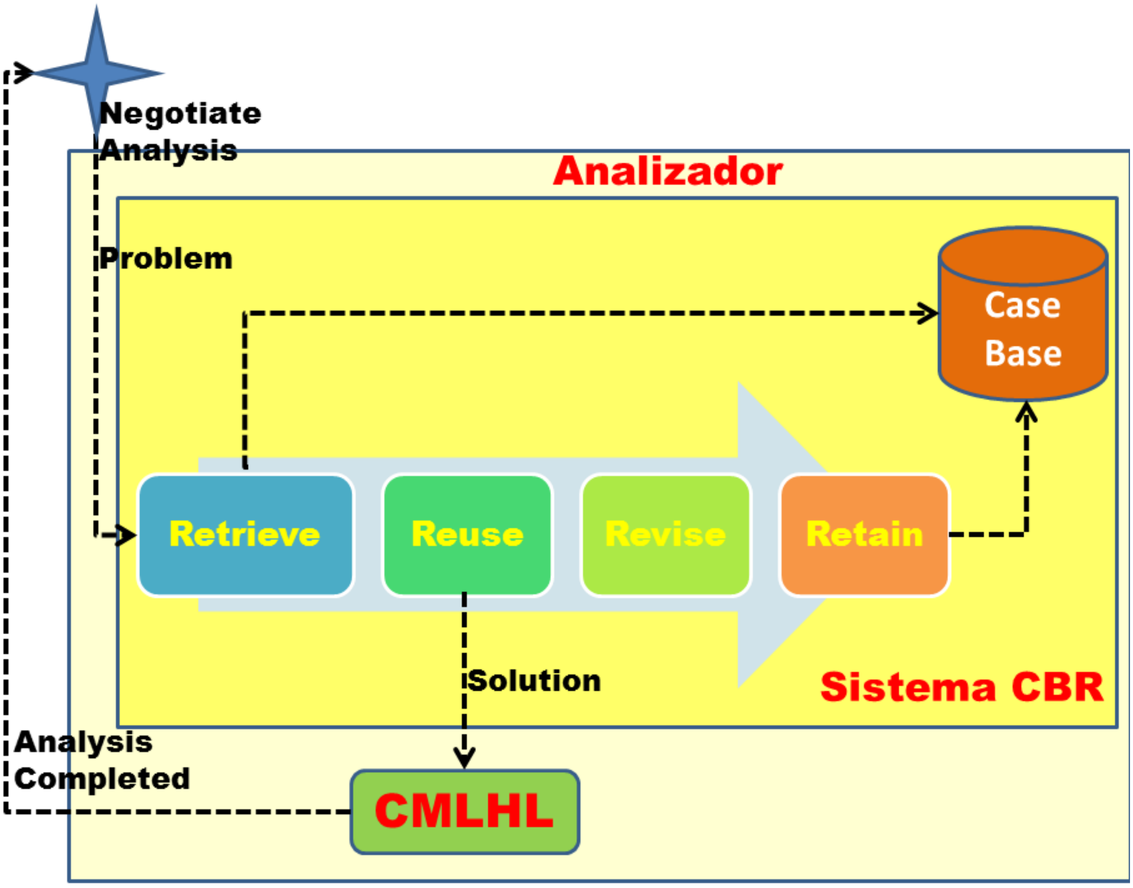


Figura 4.11 Arquitectura interna del analizador (CORCHADO, HERRERO, 2013).

Para el entrenamiento de la ANN; claro está que en las etapas del pre-procesado y el análisis de los datos, el consumo de los recursos computacionales es mayor. Por tal motivo se podría considerar por el momento una limitación, en vista que una organización dispone de diversas LANs dentro de su arquitectura de redes; tema que será considerado y tratado en futuros trabajos por el autor.

5. CAPÍTULO QUINTO

" Haz que los adversarios vean como extraordinario lo que es ordinario para ti; haz que vean como ordinario lo que es extraordinario para ti."

SUN TZU, 孫子 (REINO DE WU, SIGLO VI A.C.), EL ARTE DE LA GUERRA.

Una vez definida la propuesta del presente PFM: "modelos de inteligencia computacional aplicados al diseño e investigación de sistema de detección de intrusos"; procederemos en este último capítulo con la fase de experimentación y de análisis de los resultados de dicha experimentación, por tal motivo nos centraremos en la fase de análisis, debido a que la mencionada experimentación está igualmente orientada hacia dicha cuestión. Para las mencionadas pruebas, el autor en lo posible ha tratado de considerar un escenario real donde nuestro análisis no sólo se centrará en conocidos puertos, protocolos, direcciones IPV4, etc. sino que analizaremos todo el tráfico de datos generados por la red, en donde se han hecho un sin número de pruebas de ataques tanto internos como externos suplantando inclusive algunas IP de los equipos de la red (señuelos), pero el mayor énfasis de estas pruebas cayó en los ataques internos (localmente) en vista que con las pruebas realizadas hemos podido ver que muchos de estos ataques son más difíciles de detectar y se confunden dentro del tráfico normal de la red lo cual muchas veces pasan desapercibidas por el administrador de redes. En la séptima edición del informe anual de seguridad en entidades financieras realizado por Deloitte**[84]**, en la que han participado más de 350 entidades financieras de todo el mundo, 19 de ellas españolas y que analiza el estado de estas compañías en materia de seguridad de la información; concluyen que los "ataques internos" y "las "fugas de información" son, por tanto los problemas más detectados en las entidades. Las proyecciones de PCA, MLHL y en especial la de CMLHL permiten identificar situaciones anómalas, muchas de estas situaciones dependiendo de dónde proceda el ataque tendrán una particularidad que se diferenciará al del tráfico normal pudiendo ser la concentración de paquetes, direcciones no paralelas, el distanciamiento que tienen con el conjunto de paquetes y muchas más particularidades que se reflejaron en la experimentación y análisis nos dan indicios de situaciones anómalas, lo cual nos permite su identificación.

5.5 EXPERIMENTOS Y RESULTADOS

Para la visualización de los resultados de la fase experimental en el presente PFM se ha considerado la utilización de los segmentos iguales simples**[85]** más demostrativos (S_1, S_7, S_9 y S_{10}) los cuales representan un tipo de ataque específico y que están descritos en el apartado**[85]**. Por último, también se ha considerado un segmento acumulado (A_9) **[85]** el cual congregará el conjunto de los segmentos visualizados en el presente PFM. Todos los experimentos han sido validados haciendo uso de los algoritmos del PCA, MLHL y CMLHL.

[84] Deloitte, es la mayor firma española de servicios profesionales. Con más de 182.000 profesionales en todo el mundo prestando servicios en 4 áreas: Auditoria, consultoría, asesoramiento jurídico y tributario y corporate finance. http//www.deloitte.com
[85] Ver apartado 4.6.2

En la Figura 5.1 se puede apreciar el scatter plot generado por la PCA, enfrentando las m (número de neuronas de salida) primeras componentes (columnas) del conjunto de datos entre sí. Dicha visualización representa el tráfico normal de la red por un tiempo de 10 minutos.

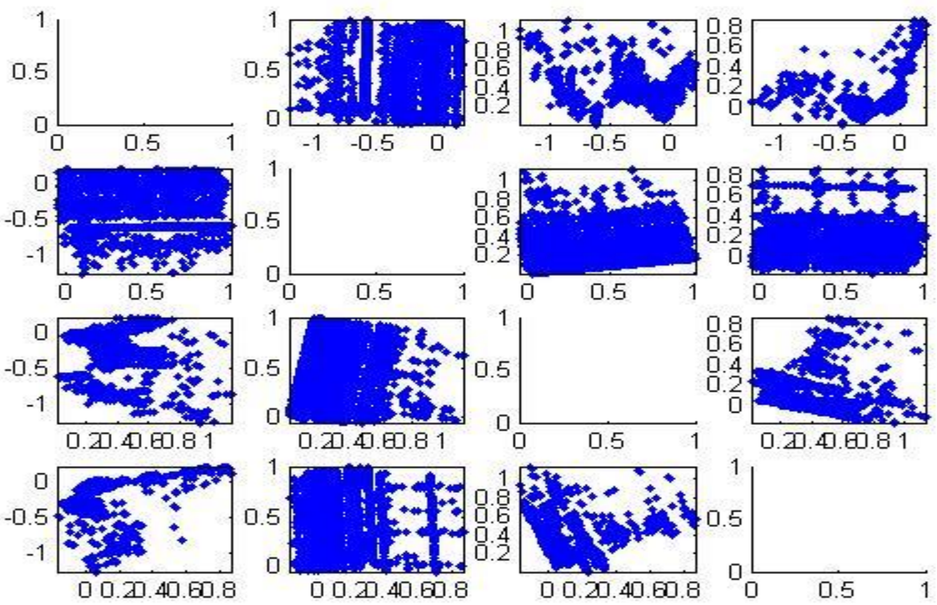


Figura 5.1 Scatter Plot, del tráfico normal.

En la Figura 5.2 apreciamos el tráfico normal de la red en 3 dimensiones.

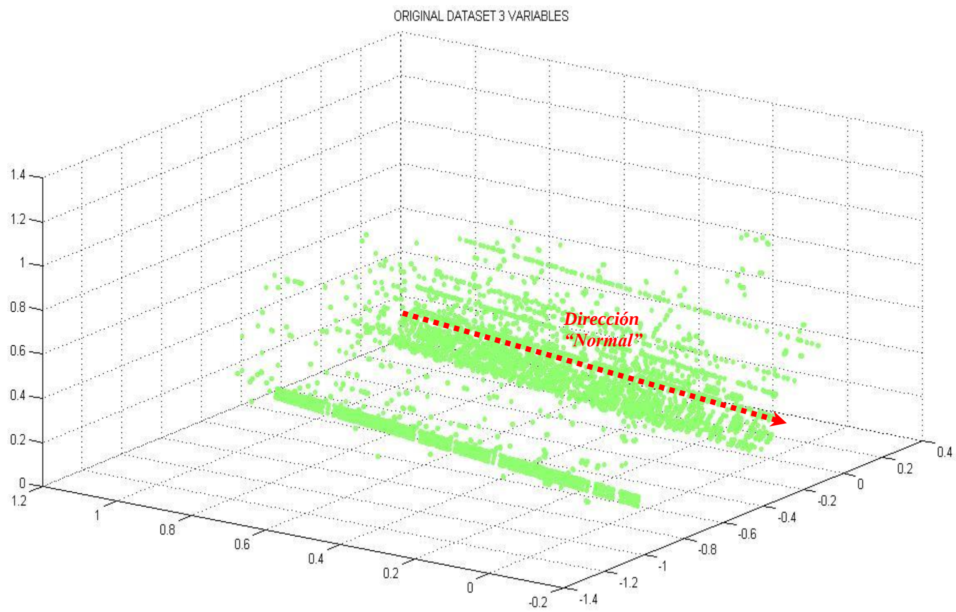


Figura 5.2 Tráfico normal de la red en 3D.

En la Figura 5.3 apreciamos la proyección PCA en 2 dimensiones, del segmento simple(S_1) conteniendo situaciones no anómalas (tráfico normal de la red)

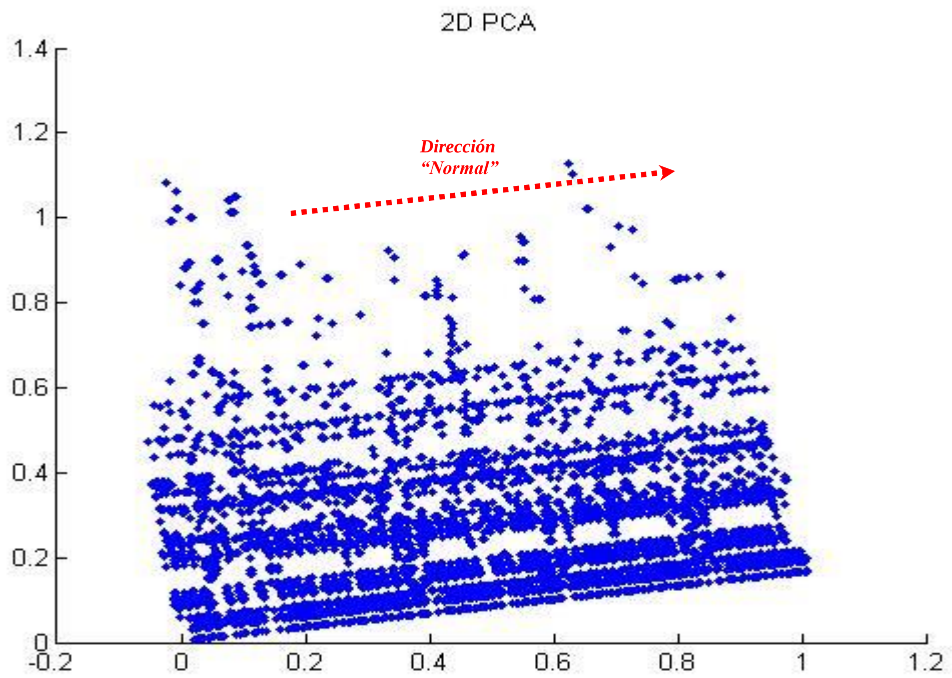


Figura 5.3 Visualización del Segmento simple-S_1.(PCA)

Parameter Values of Figura 5.3

Parameter	Value
Variables de Salida	4
Iterations	-
Learning rate	-
P	-
Tau	-

Tabla 5.1 Parámetros del S_1 (PCA)

En la Figura 5.4 podemos apreciar la proyección que nos brinda el MLHL

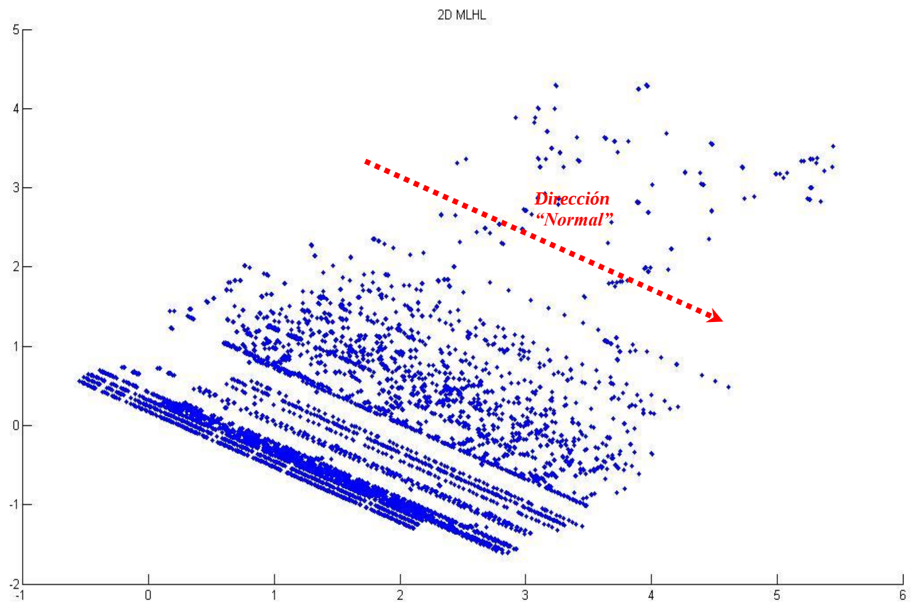


Figura 5.4 Visualización del Segmento simple-S_1. (MLHL)

Parameter Values of Figura 5.4

Parameter	Value
# Neuronas (m)	4
Iterations	100000
Learning rate	0.025
P	0.025
Tau	-

Tabla 5.2 Parámetros del S_1 (MLHL)

En la Figura 5.5 podemos apreciar la proyección que nos brinda el CMLHL.

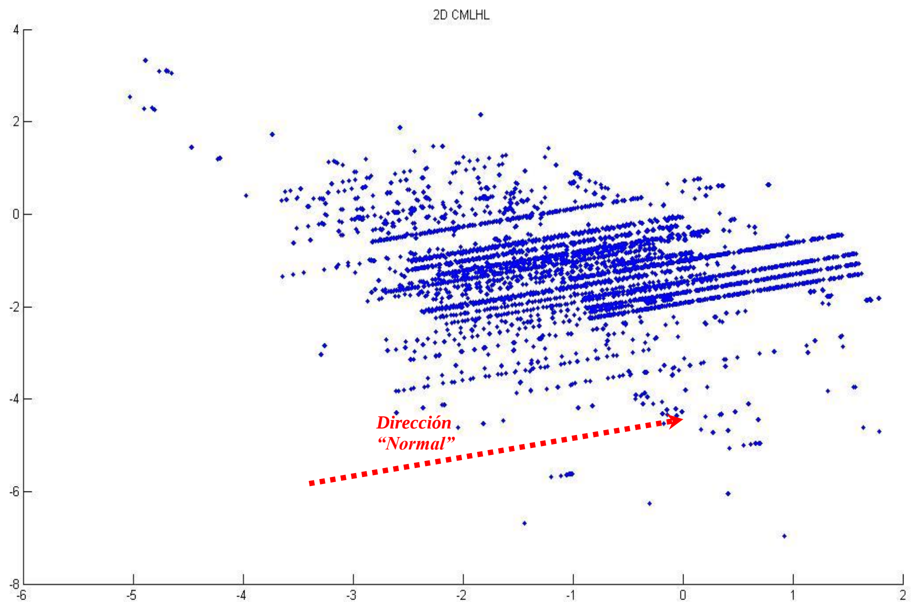


Figura 5.5 Visualización del Segmento simple-S_1. (CMHL)

Parameter Values of Figura 5.5

Parameter	Value
# Neuronas (m)	4
Iterations	100000
Learning rate	0.025
P	0.027
Tau	0.13

Tabla 5.3 Parámetros del S_1 (CMLHL)

A continuación visualizaremos el segmento simple (S_7) donde se originó una situación anómala en la red, en total tienen una duración de 20 minutos aproximadamente. En estos segmentos la prueba consistió en penetrar el sistema, atacando el puerto 22 (servicio ssh) donde el atacante con IP 192.168.1.36 obtiene permisos de “Root” accediendo remotamente al servidor para luego hacer ping a la IP 173.194.41.24 (Google). Apreciaremos la proyección 2D del PCA, MLHL y CMLHL respectivamente. El tamaño de los paquetes, que son diferenciados con el color rojo (S_6) y verde (S_7), del tráfico normal; nos da muestra de cómo dicho ataque que se realizó localmente se confunden con la dirección del tráfico normal de la red, pero el detalle es que el tamaño de dichos paquetes está ubicado en una zona distinta y es diferente con el conjunto de paquetes que se encuentran agrupados.

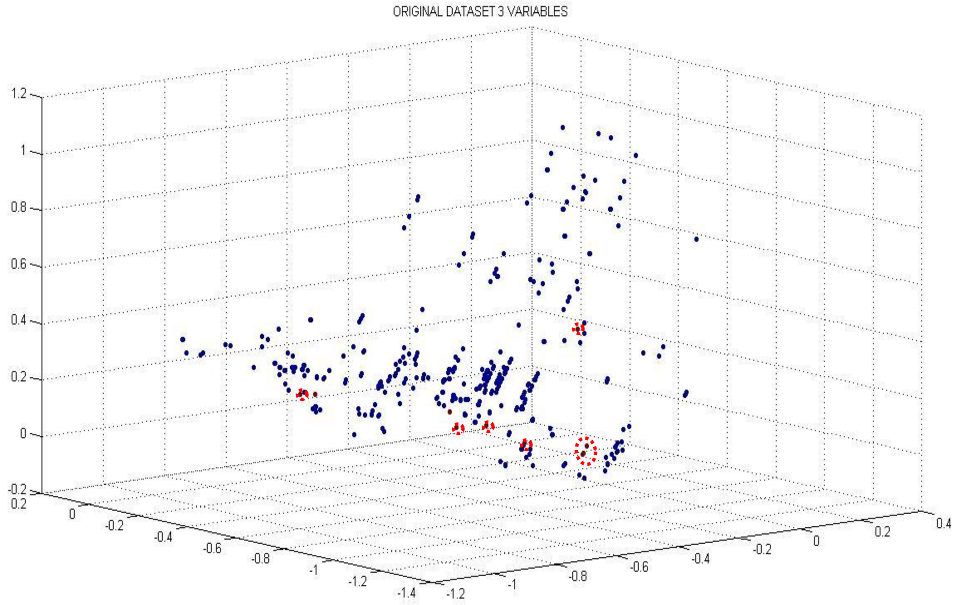


Figura 5.6 Visualización del Segmento simple-S_7. (PCA)

Parameter Values of Figura 5.6

Parameter	Value
Variables de Salida	4
Iterations	-
Learning rate	-
P	-
Tau	-

Tabla 5.4 Parámetros del S_7 (PCA)

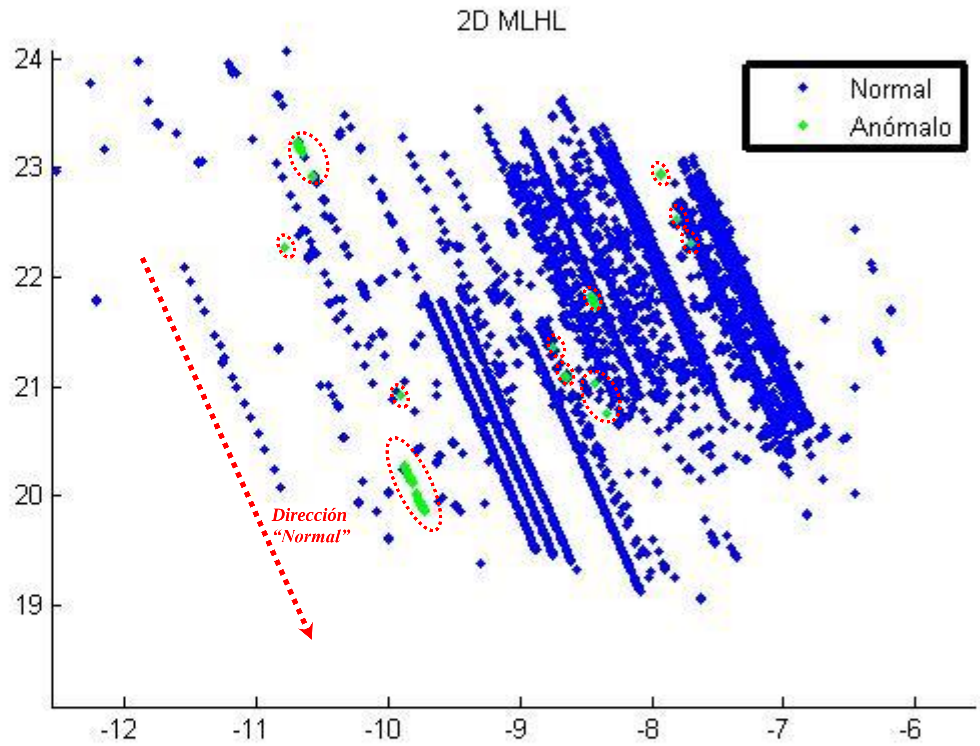


Figura 5.7 Visualización del Segmento simple-S_7. (MLHL)

Parameter Values of Figura 5.7	
Parameter	**Value**
# Neuronas (m)	4
Iterations	100000
Learning rate	0.031
P	0.08
Tau	-

Tabla 5.5 Parámetros del S_7 (MLHL)

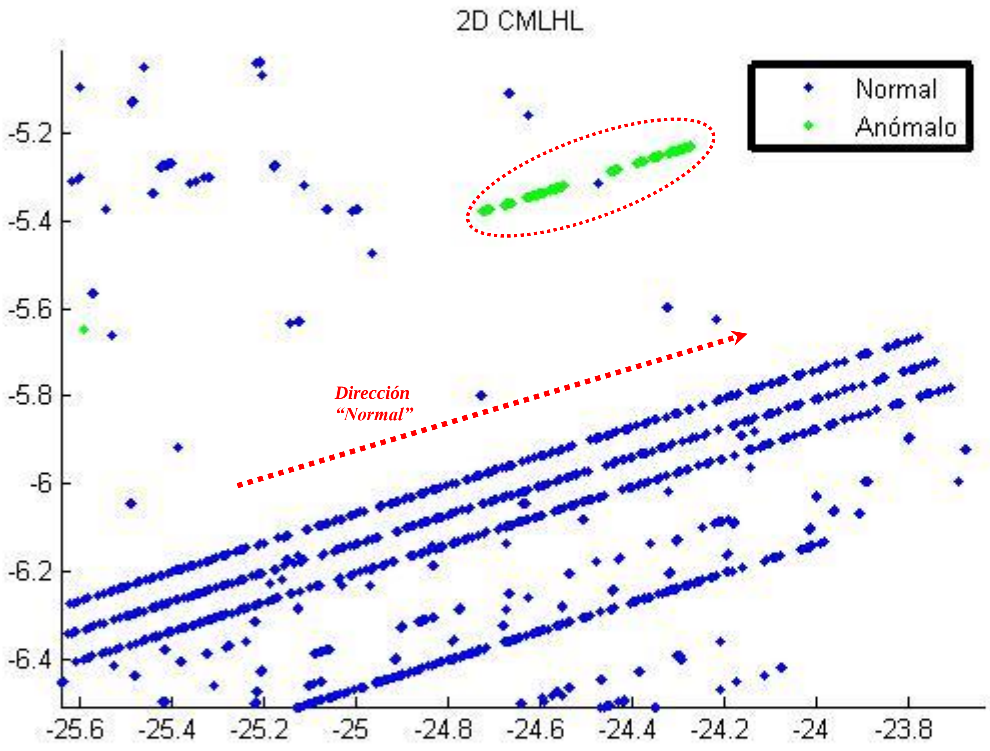


Figura 5.8 Visualización del Segmento simple-S_7. (CMLHL)

Parameter Values of Figura 5.8

Parameter	Value
# Neuronas (m)	4
Iterations	100000
Learning rate	0.031
P	0.085
Tau	0.085

Tabla 5.6 Parámetros del S_7 (CMLHL)

A continuación visualizaremos el segmento simple (S_8), en total tienen una duración de 10 minutos aproximadamente, donde el tipo de ataque fue una denegación de servicio que consistió en la inundación de paquetes TCP-SYN flags por un tiempo de sólo 6 segundos, más detalle del ataque se encuentra descrito en el ANEXO F. Apreciaremos la proyección 2D del PCA, MLHL y CMLHL respectivamente. El tamaño de los paquetes, que son diferenciados con el color cian (S_8) del tráfico normal; vemos que toman un direccionamiento continuo contrario al del tráfico normal y es debido a que dicho ataque es del tipo spoofing en la cual la identidad del atacante es suplantada por un sin número de direcciones IPs (no locales).

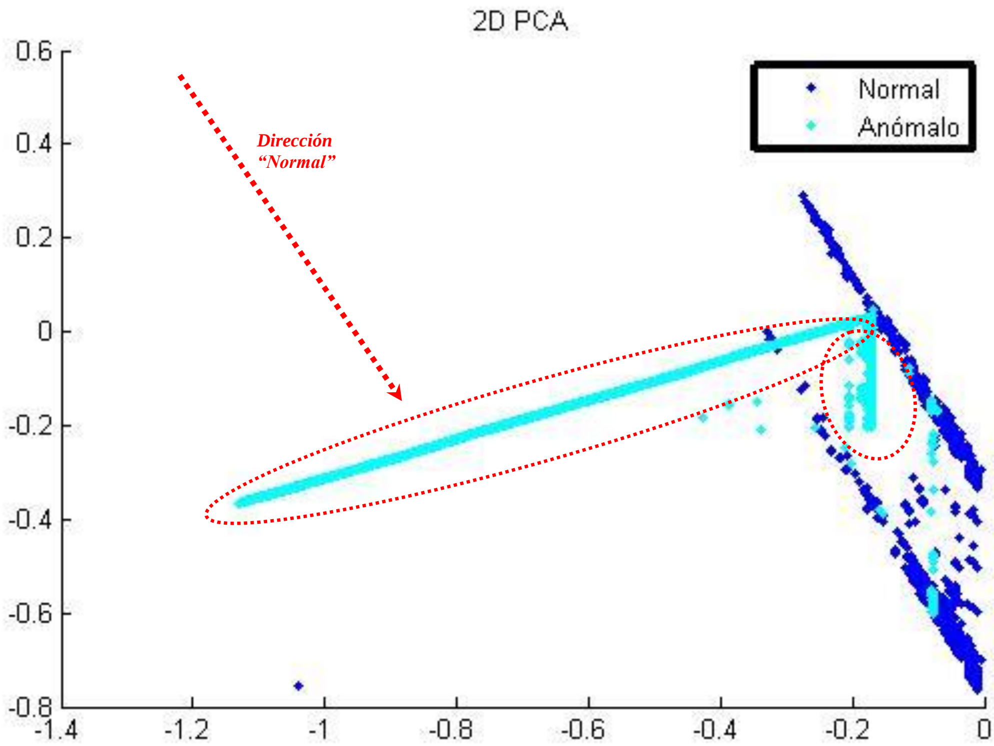


Figura 5.9 Visualización del Segmento simple-S_8 (PCA)

Parameter Values of Figura 5.9

Parameter	Value
Variables de Salida	4
Iterations	-
Learning rate	-
P	-
Tau	-

Tabla 5.7 Parámetros del S_8 (PCA)

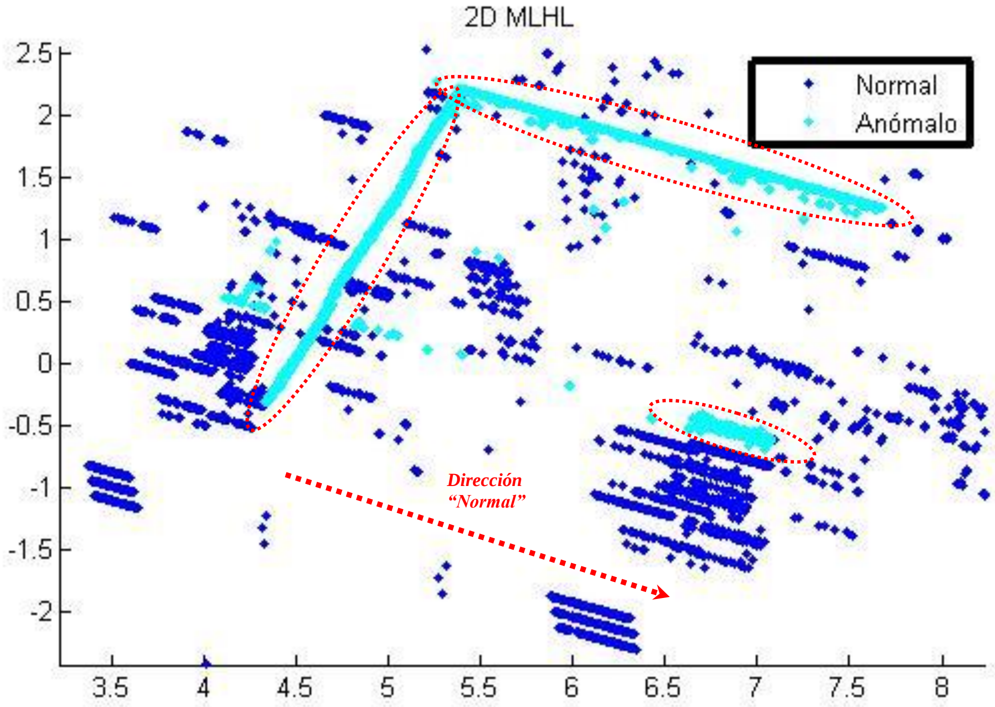


Figura 5.10 Visualización del Segmento simple-S_8 (MLHL)

Parameter Values of Figura 5.10

Parameter	Value
# Neuronas (m)	4
Iterations	100000
Learning rate	0.031
P	0.08
Tau	-

Tabla 5.8 Parámetros del S_8 (MLHL)

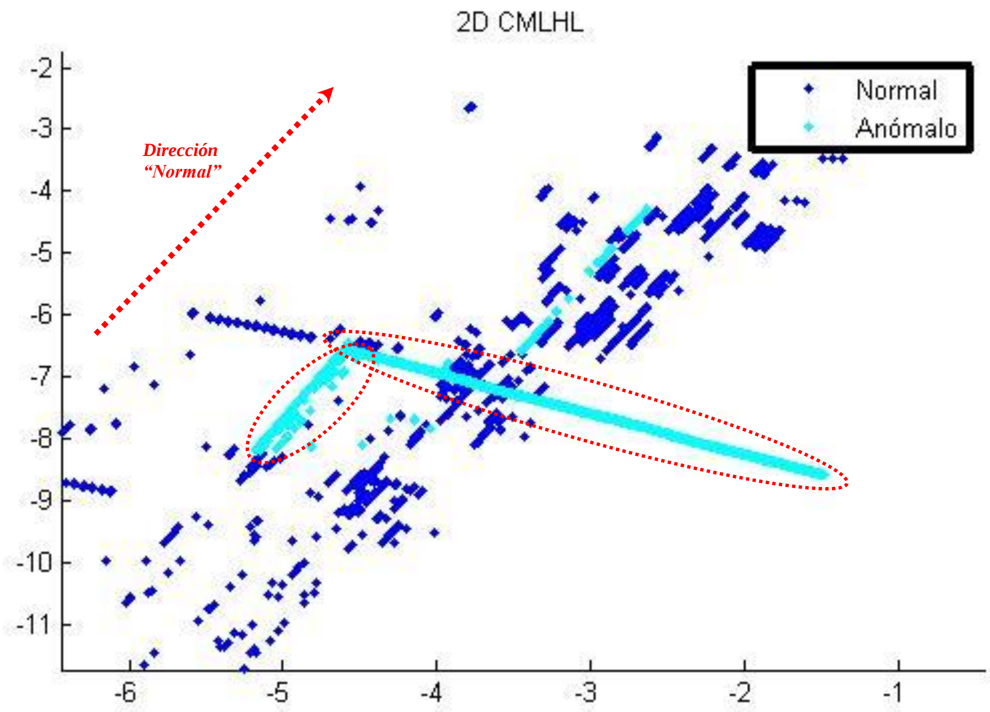


Figura 5.11 Visualización del Segmento simple-S_8. (CMLHL)

Parameter Values of Figura 5.11

Parameter	Value
# Neuronas (m)	4
Iterations	100000
Learning rate	0.031
P	0.08
Tau	0.12

Tabla 5.9 Parámetros del S_8 (CMLHL)

A continuación visualizaremos el segmento simple (S_9), en total tienen una duración de 10 minutos aproximadamente, donde el tipo de ataque fue una denegación de servicio que consistió en la inundación de paquetes UDP flags por un tiempo de sólo 8 segundos, más detalle del ataque se encuentra descrito en el ANEXO F. Apreciaremos la proyección 2D del PCA, MLHL y CMLHL respectivamente. El tamaño de los paquetes, que son diferenciados con el color magenta (S_9) del tráfico normal; vemos que toman un direccionamiento continuo contrario al del tráfico normal y es debido a que dicho ataque es del tipo spoofing en la cual la identidad del atacante es suplantada por un sin número de direcciones IPs (direcciones no locales).

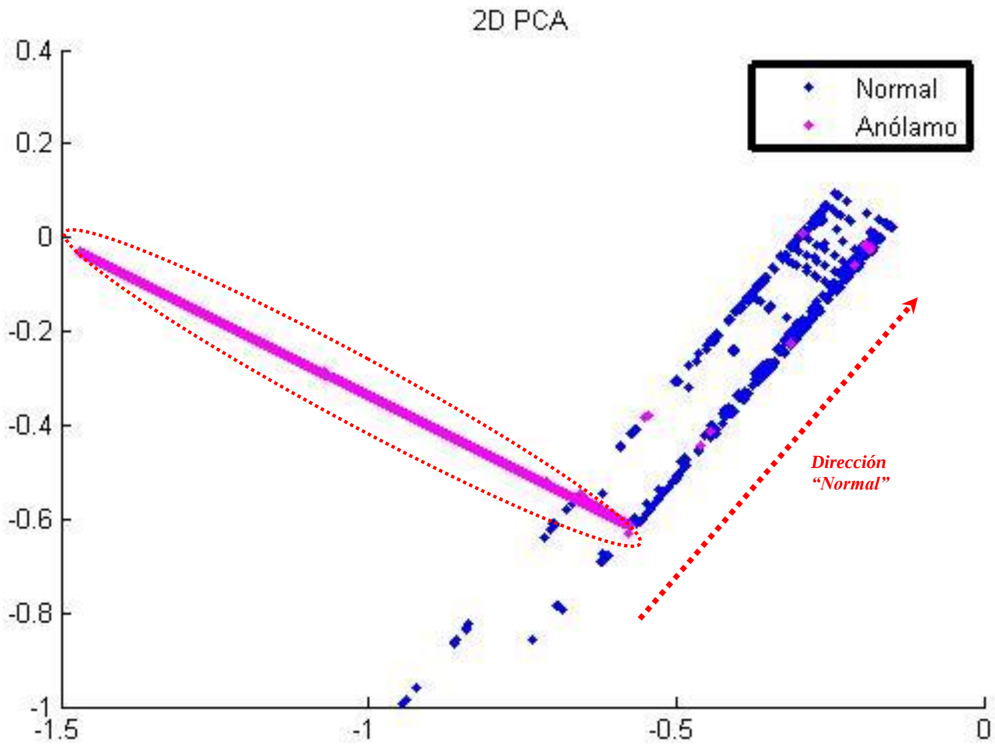


Figura 5.12 Visualización del Segmento simple-S_9. (PCA)

Parameter Values of Figura 5.12

Parameter	Value
Variables de Salida	4
Iterations	-
Learning rate	-
P	-
Tau	-

Tabla 5.10 Parámetros del S_9 (PCA)

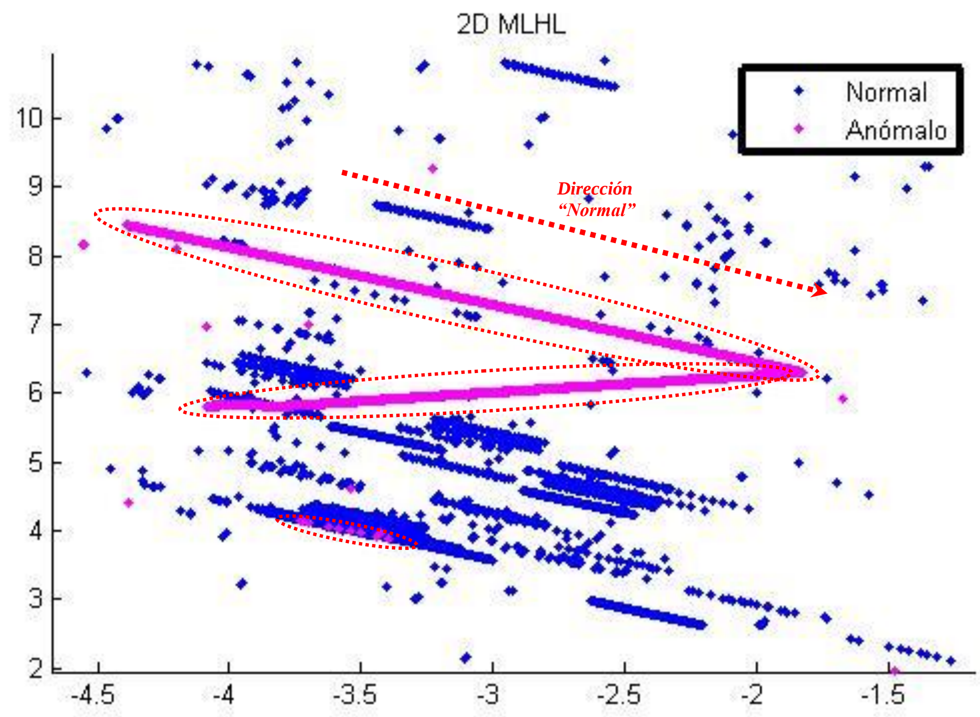


Figura 5.13 Visualización del Segmento simple-S_9. (MLHL)

Parameter Values of Figura 5.13

Parameter	Value
# Neuronas (m)	4
Iterations	100000
Learning rate	0.028
P	0.09
Tau	-

Tabla 5.11 Parámetros del S_9 (MLHL)

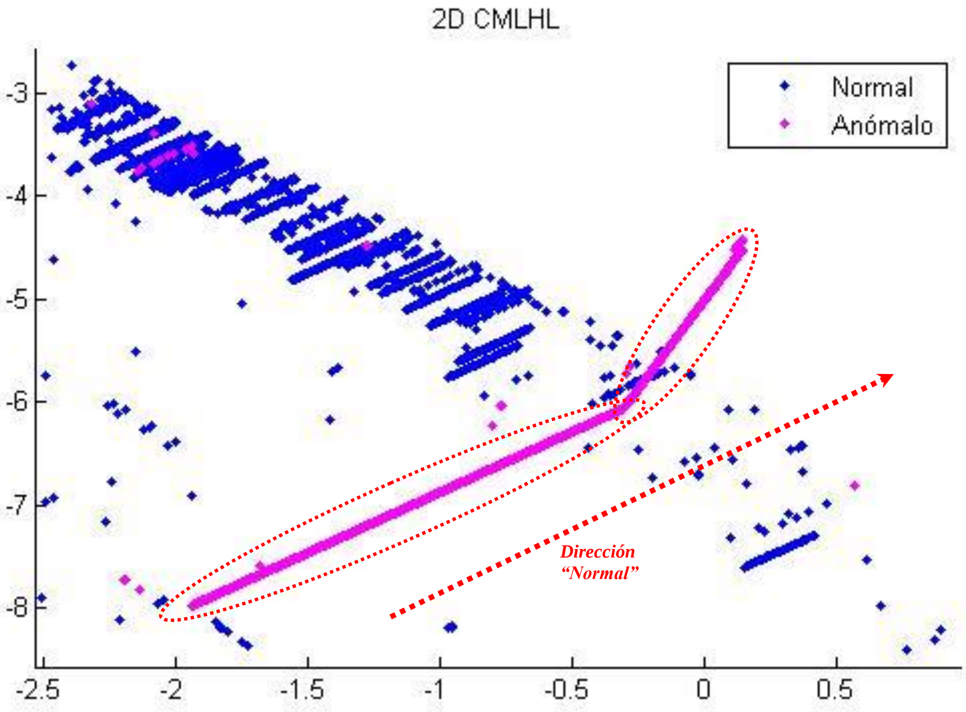


Figura 5.14 Visualización del Segmento simple-S_9. (CMLHL)

Parameter Values of Figura 5.14

Parameter	Value
# Neuronas (m)	4
Iterations	100000
Learning rate	0.028
P	0.085
Tau	0.11

Tabla 5.12 Parámetros del S_9 (CMLHL)

Por último visualizaremos el segmento aumentado (A_9), en total tienen una duración aproximada de 50 minutos de tráfico de red, donde agruparemos todas las pruebas realizadas anteriormente en los segmentos iguales simples (S_1,S_7,S_8 y S_9) para esto hemos tenido que eliminar los overlaps existentes entre cada dos segmentos iguales consecutivos, cuya duración era de aproximadamente 2 minutos. Apreciaremos la proyección 2D del PCA, MLHL y CMLHL respectivamente.

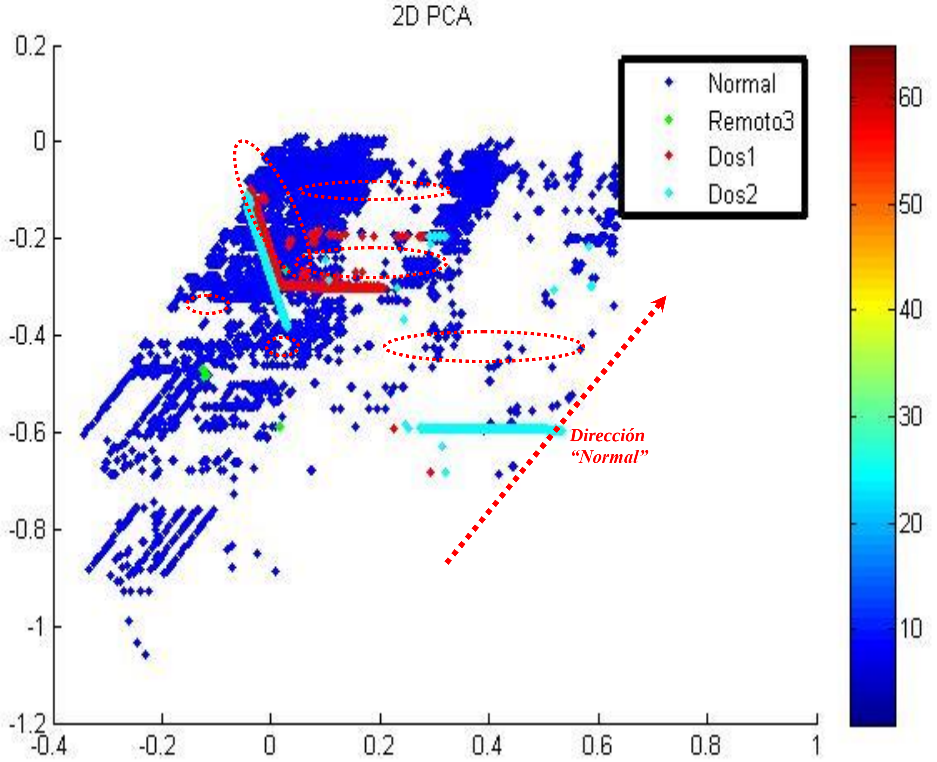


Figura 5.15 Visualización del Segmento aumentado-A$_9$. (PCA)

Parameter Values of Figura 5.15

Parameter	Value
Variables de Salida	4
Iterations	-
Learning rate	-
P	-
Tau	-

Tabla 5.13 Parámetros del A$_9$ (PCA)

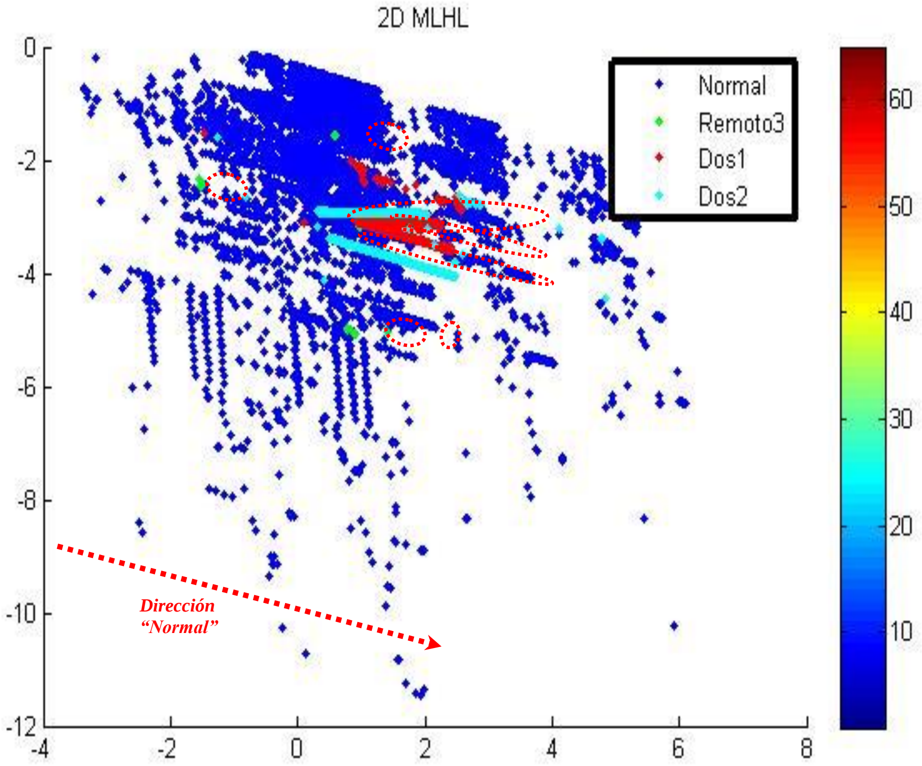


Figura 5.16 Visualización del Segmento aumentado-A_9. (MLHL)

Parameter Values of Figura 5.16	
Parameter	**Value**
# Neuronas (m)	4
Iterations	100000
Learning rate	0.029
P	0.08
Tau	-

Tabla 5.14 Parámetros del A_9 (MLHL)

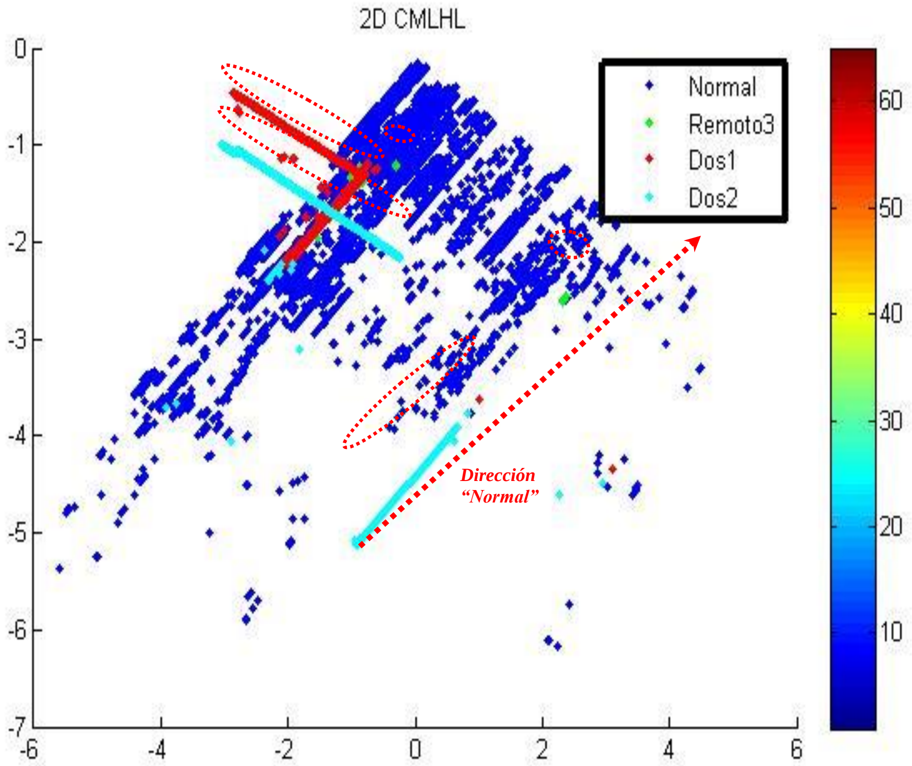


Figura 5.17 Visualización del Segmento aumentado-A_9. (CMLHL)

Parameter Values of Figura 5.17

Parameter	Value
# Neuronas (m)	4
Iterations	100000
Learning rate	0.029
P	0.09
Tau	0.09

Tabla 5.14 Parámetros del A_9 (CMLHL)

6. CAPÍTULO SEXTO

"The reward of a thing well done is to have done it."

RALPH WALDO EMERSON (BOSTON, 1803-1882).

Habiendo examinado las proyecciones obtenidas en los experimentos mostrados en el Capítulo Quinto, se puede decir que en el presente PFM se ha podido representar el tráfico normal de un modo compacto e intuitivo facilitando de esta forma la tarea de monitorización de la red de un sólo vistazo. Además de permitirnos el identificar aquellas interrupciones de tráfico en la red generados por actividades anómalas muchas de ellas asociadas como consecuencias de ataques o disfuncionamientos.

De todos los experimentos presentados en el Capítulo Quinto (Figura 5.6 al 5.17), se ha podido ver que los ataques realizados localmente muchas veces se confunden con el del tráfico normal, pero la dirección, la dimensión y la localización de los paquetes las distinguen de lo normal. Así mismo aquellos ataques realizados desde el exterior de nuestra red son representados como líneas no paralelas a la dirección en la cual el tráfico normal se desarrolla. Por lo tanto no es fácil el establecer exactamente que ocurre en un momento dado; es en este preciso instante donde el análisis cumple un papel sumamente importante por tal motivo nos apoyamos en uno de los paradigmas del presente PFM que es el estudio de casos (CBR), donde el análisis incorpora dos comportamientos diferentes que son: “aprendiendo” y la “explotación”, lo cual fue tratado más en detalle en el Capítulo Cuarto, apartado 4.6.3.

Los resultados del estudio de pruebas demuestran la capacidad de generalización, porque muestra "el modelo visual" de ataques nuevos incluso si el modelo neuronal no ha afrontado tales ataques antes. Debería ser notado que los atacantes emplean estrategias muy diferentes para poder pasar inadvertidos en la red haciendo uso de diversas técnicas como: la extensión de los paquetes de ataque con el tiempo o reduciendo la cantidad de paquetes en un ataque, el empleo de señuelos, etc; harán que se visualicen menos claramente dichos ataques, pero con el empleo del CMLHL se pueden identificar dichos ataques con mayor efectividad como ha sido mostrado en las figuras 5.8, 5.11, 5.14 y 5.17.

Comparando CMLHL con otros modelos no supervisados (apartado 3.1.4.2), podemos concluir que PCA y MLHL en la identificación de situaciones anómalas son menos eficientes y eficaces. Por tal motivo CMLHL será el modelo a ser utilizado para próximos proyectos de investigación relacionados al campo de ID.

6.1 CONCLUSIONES

Los métodos de proyección de grandes cantidades de datos a través de técnicas que puedan brindarnos la visualización de dichos datos en diversas dimensiones es sumamente fascinante pero a su vez complejo. Es así como los datos son proyectados en un subespacio inferior dimensional trazado en dos o tres

dimensiones haciendo posible el examinar la estructura a simple vista, brindándonos de esta forma una mayor facilidad para su análisis. Así esto proveerá al administrador de la red o al encargado de la seguridad de un snapshot general del tráfico de la red (interacciones de protocolos, volumen de tráfico, la comunicación entre los hosts, etc.) para que así ellos puedan identificar aquellas situaciones anómalas en la red. Los instrumentos de visualización tienen que ser diseñados de modo que las anomalías fácilmente puedan ser señaladas para el análisis posterior por analistas más experimentados. Por lo tanto podemos decir que las herramientas de visualización deben contribuir a la tarea de la seguridad de la manera siguiente:

- Deben ser intuitivos y requerir de un tiempo reducido de configuración.
- El proveer una visualización intuitiva del tráfico de la red permite que la tarea del monitoreo de la red puede ser asignado al personal menos experimentado, reduciendo la carga de trabajo al personal de mayor experiencia.
- Las visualizaciones pueden ayudar al administrador de la red a prepararse para acontecimientos imprevistos, pudiendo ser usados en la educación de la seguridad.
- El poder trabajar en armonía con otros instrumentos para la ID de un modo complementario.

La faceta interesante del aprendizaje que cumplen las ANN no es solamente que el modelo de entrada puede ser "aprendido/clasificado/identificado" con precisión, sino que este aprendizaje puede ser generalizado, mientras el aprendizaje toma lugar sobre un modelo de entrenamiento, una importante propiedad en el proceso de aprendizaje es que la red puede generalizar estos resultados sobre un conjunto de modelos que no fueron previamente aprendidos. En la identificación de modelos desconocidos es donde encajan los Zero Day Attacks.

Considerando lo mencionado en líneas anteriores, el diseño del sistema, el empleo de diferentes paradigmas de AI, el modelamiento del presente PFM nos brinda las siguientes ventajas:

- Capacidad de generalización, haciendo uso del modelo CMLHL, ataques nunca antes vistos pueden ser identificados.
- Escalabilidad, facilidad del cambio y/o configuración sin perder la calidad para adaptarse a las circunstancias cambiantes.
- Tolerancia de fallos, disponiendo de la información en todo momento, mostrándonos proactivos.
- Optimización de los recursos, gracias a la arquitectura del CBR se pueden optimizar recursos computacionales para el análisis de los datos.

- Tiempo de ejecución de procesamiento, con una adecuada localización de agentes analizadores en diferentes puntos de la red.

- Visualización, pudiendo ser realizada esta tarea en una amplia variedad de dispositivos.

  Así mismo algunas limitaciones presentadas en el modelo son:

- Prescindir del factor humano, como está basado en el aprendizaje no supervisado, automáticamente no puede levantar alarmas advirtiéndonos de los ataques. Requiriendo una constante supervisión humana.

- Alto consumo de recursos computacionales, debido a la gran cantidad de información extraída de la captura del tráfico de la red, se necesita de recursos computacionales elevados para el preprocesado y el análisis de los datos.

- Seguridad en los datos, las técnicas utilizadas para la extracción y manipulación de los datos del tráfico de la red, deben estar sumamente protegidos o apartados a una red privada; porque de ser detectados por un Hacker , éste tratará de modificar y/o eliminar todo tipo de rastros causados en su intrusión al sistema.

La Detección de Anomalías se convierte en la opción más interesante, y que presenta a priori mayores posibilidades. De hecho, precisamente es en esta área en el que se concentra actualmente el mayor número de investigaciones. Sin embargo, tal y como se ha explicado en los capítulos anteriores, dicha filosofía de análisis presenta importantes inconvenientes, como pueden ser el exceso de falsos positivos o la propensión a la inestabilidad de sus modelos de representación de conocimiento. Por ello, es necesario desarrollar nuevos métodos de análisis que integren de forma homogénea las capacidades de los métodos de Detección de Anomalías y de los de Detección de Usos Indebidos, potenciando de forma sinérgica las ventajas de ambos, y minimizando sus desventajas. De esta forma, será posible responder adecuadamente ante ataques bien conocidos, a la vez que ante ataques de nuevo diseño.

Finalmente la propuesta del presente PFM vendría a ser un complemento a otras herramientas de software o hardware para la seguridad de las redes del sistema. Muchas organizaciones vienen utilizando dichas herramientas de seguridad (Firewalls, IDS, IPS, etc.), así mismo el modelamiento propuesto en el PFM puede trabajar en un ambiente de armonía, pudiéndolo utilizar en paralelo sobre la misma fuente de datos.

6.2 TRABAJOS FUTUROS

A raíz de los estudios llevados a cabo durante el desarrollo del presente PFM es posible señalar varias temáticas susceptibles a aportar valor añadido a los

actuales métodos de ID y sobre todo a lo propuesto en el presente PFM, los cuales son:

- Mejora en el tratamiento de los datos, elaborando un diseño capaz de automatizar dicho proceso.

- El empleo de técnicas de visualización aún tienen que ser mejoradas, llegando al punto de disponer de diferentes técnicas de visualización a la vez.

- Mejoramiento en la arquitectura del sistema para su protección ante intrusiones y/o ataques maliciosos.

- Optimización del modelo mediante el análisis de sensibilidad, manera que se eliminen de él parámetros de detección no suficientemente representativos, logrando así la optimización de los tiempos de respuesta.

“In the future, how we gather, manage and use information will determine whether we succeed or fail”. B. Gates

Bibliografía

ALESSANDRI, d. , Cachin C., Dacier M., Deak O., Julisch K., Randell B., Riordan J., Tscharner A., Wespi A. y Wüest C. (2001) **“Towards a Taxonomy of Intrusion Detection Systems and Attacks”**. Research Report RZ 3366, IBM Research, Zurich Research Laboratory, Suiza, Septiembre 2001.

AHLBERG C., Shneiderman, B. (1999) **“Visual Information Seeking: Tight Coupling of Dy-namic Query Filters with Starfield Displays”**. (ed) Readings in Information Visualization: using Vision to Think. Morgan Kaufmann Publishers Inc. 244-250

ALLEN J., Christie A., Fithen W., McHugh J., Pickel J. y Stoner E.(2000) **“State of the Practice of Intrusion Detection Technologies”**. Carnegie Mellon University Software Engineering Institute, Technical Report, 2000.

AXELSSON S. **“Intrusion Detection Systems: A Survey and Taxonomy”**. Technical report 99-15, Department of Computer Engineering, Chalmers University. Göteborg, Suecia, Marzo 2000.

BALEPIN I., Maltsev, S., Rowe, J., Levitt, K. (2003) “**Using Specification-Based Intrusion Detection for Automated Response”.** (ed) Sixth International Symposium on Recent Ad-vances in Intrusion Detection (RAID 2003) 2820. Springer, Heiderlberg. LNCS

BACE R. y Mell P. **“Intrusion Detection Systems”**. NIST Special Publication on Intrusion Detection Systems. Disponible en http://csrc.nist.gov/publications/nistpubs/, 2001.

BASTOS, Eduardo. **“Tópicos em Segurança de Redes”**. In: Seminário Internacional de Educação Tecnológica. Novo Hamburgo, 2004. Fundação Escola Técnica Liberato, 2004.

BECKER, R.A., Eick, S.G., Wilks, A.R. (1995) **“Visualizing Network Data”**. IEEE Transac-tions on Visualization and Computer Graphics 1(1): 16-28

BOE, 2003. Artículo 6, **“Ley 59/2003, de 19 de diciembre, de firma electrónica, BOE”.** España, 2003. http://www.boe.es/buscar/act.php?id=BOE-A-2003-23399&p=20071229&tn=1#a6

BORGHELLO, Cristian. Tesis: **“Seguridad Informática : Sus Implicancias e Implementaciones”**, 2001. http://www.segu-info.com.ar/

BRATMAN, M.E. (1987). **“Intentions, Plans and Practical Reason”**. Harvard University Press, Cambridge, M.A.

CARRASCOSA, C., Bajo, J., Julián, V., Corchado, J.M., Botti, V. (2008). **“Hybrid Multi-agent Architecture as a Real-Time Problem-Solving Model”.** Expert Systems with Applications: An International Journal 34(1): 2-17

CASWELL, Brian et al**. “Snort 2: Sistema de detecção de intruso”**. Rio de Janeiro: Altas Books, 2003.

CERT/IN03, CERT/In. Indian Computer Emergency Response Team. **“IDS- Intrusion Detection System”**. CERT/In Security Guideline CISG-2003-06, 2003.

CHIRAG Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel, Muttukrishnan Rajarajan. **“A survey of intrusion detection techniques in Cloud”.** Review Article. *Journal of Network and Computer Applications*, *Volume 36, Issue 1*, *January 2013*, *Pages 42-57*

CHOI, H., Lee, H., Kim, H. (2009**) “Fast Detection and Visualization of Network Attacks on Parallel Coordinates”**. Computers & Security 28(5): 276-288

CHRISTOPHER Day. **Chapter 26 – “Intrusion Prevention and Detection Systems *Computer and Information Security Handbook”*** *(Second Edition)*, *2013*, *Pages 485-498*

CMT, tercer trimestre 2013, **“Informe e-commerce”.** Comisión del Mercado de Telecomunicaciones.España.
http://www.cmt.es/c/document_library/get_file?uuid=f33f361d-9126-4f40-8afb-88b430b98150&groupId=10138

CONTI, G. (2007**) "Security Data Visualization: Graphical Techniques for Network Analy-sis"**. No Starch Press

CORCHADO, Emilio; HERRERO, Álvaro. **RT-MOVICAB-IDS: Addressing real-time intrusion detection**. Spain, 2013. University of Salamanca.

CORCHADO Emilio, KROMER Pavel, PLATOS Jan, SNÁSEL Vaclav, Garcia Laura. (2010) "**Neural PCA and Maximum Likelihood Hebbian Learning on the GPU".** Original Research Article. This research is partially supported through a projects of the Spanish Ministry of Economy and Competitiveness [ref: TIN2010-21272-C02- 01] (funded by the European Regional Development Fund). This work was also support-ed by the European Regional Development Fund in the IT4Innovations Centre of Ex-cellence project (CZ.1.05/1.1.00/02.0070) and by the Bio-Inspired Methods: re-search, development and knowledge transfer project, reg. no.CZ.1.07/2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state.

CORCHADO Emilio, HERRERO Álvaro, GASTALDO Paolo, ZUNINO Rodolfo. (2009) **"Neural projection techniques for the visul inspection of network traffic"** Original Research Article. *Applied NeuroComputing, 21 de junio* 2009. http://www.sciencedirect.com/science/article/pii/S0925231209002082

CORCHADO Emilio, HERRERO Álvaro. (2011) **"Neural visualization of network traffic data for intrusion detection"** Original Research Article. *Applied Soft Computing*, *Volume 11, Issue 2*, *March 2011*, *Pages 2042-2056.*

CORCHADO Emilio, HERRERO Álvaro, PINZON Cristian, PAZ Juan, BAJO Javier , CORCHADO Juan. (2011) . **"idMAS-SQL:Intrusion detection base on MAS to detect and block SQL injection through data mining"** . Original Research Article. Information Sciences, 6 de Julio 2011.

CSAIL, MIT **"Computer science and artificial intelligence laboratory"**, http://www.csail.mit.edu/

CURIEL, Leticia. **"SIBIC: Sistema de Identificación basado en Inteligencia Computacional"**. Tesis Doctoral. Director: Emilio Corchado. Universidad de Burgos, 2010.

DAEJOON Joo, TAEHO Hong, INGOO Han."**The neural network models for IDS based on the asymmetric costs of false negative errors and false positive errors"** Original Research Article. *Expert Systems with Applications*, *Volume 25, Issue 1*, *July 2003*, *Pages 69-75.*

DASGUPTA, D., Gonzalez, F., Yallapu, K., Gomez, J., Yarramsettii, R. (2005) CIDS: "**An Agent-based Intrusion Detection System"**. Computers & Security 24(5): 387-398

D'AMICO, A.D., Goodall, J.R., Tesone, D.R., Kopylec, J.K. (2007) **"Visual Discovery**

in Computer Network Defense”. IEEE Computer Graphics and Applications 27(5): 20-27

DEBAR H., Dacier M. y Wespi A. **“Towards a Taxonomy of Intrusion-Detection Systems”**. Computer Networks, Vol. 31, No. 8, Pag. 805-822, Abril 1999.

DECSAI, **“Departamento de Ciencias de la computación e I.A”**. Universidad de Granada. http://decsai.ugr.es/

DE LIMA, Igor Vinícius M. “**Uma abordagem simplificada de detecção de intrusão baseada em redes neurais artificiais”**. Florianópolis, 2005. (Mestrado Ciências da Computação) Universidade Federal de Santa Catarina, 2005.

DURSTELER, Juan Carlos. **“Coordenadas Paralelas”.** http://www.infovis.net/printMag.php?num=201&lang=1 . Marzo 2010.

DURSTELER, Juan Carlos. **“City’o’Scope”.** http://www.infovis.net/printMag.php?num=54&lang=1. Agosto 2001.

ENISA, 2002. Cinta Castillo Jiménez. Universidad de Sevilla. España. http://www.juntadeandalucia.es/institutodeadministracionpublica/anuario/articulos/descargas/01_EST_02_castillo.pdf

ESET, 2012. Operación Medre: espionaje industrial en Latinoamérica. http://www.eset-la.com/pdf/prensa/informe/operacion_medre.pdf

FERNANDES, Anita Maria da Rocha. “**Inteligência Artificial – Noções Gerais”**. Florianópolis: VisualBooks, 2005.

FRANCESQUINI, Emilio de Camargo. **“Técnicas de detecção de intrusos para redes móveis sem fio”.** São Paulo: USP, 2004. Disponível em: http://grenoble.ime.usp.br/movel/tecnicasIntruso.pdf

GARCÍA, Pablo. **“Entorno de Seguridad Inteligente para la Detección y Prevención de Intrusiones de Red basado en la detección unificada de patrones y anomalías”.** Tesis Doctoral. Universidad de Deusto. España, julio 2006.

GOLOMB G. **“Intrusion Detection Methodologies Demystified”**. Enterasys Networks Incorporated Technical Whitepaper, Diciembre 2003.

GONZÁLEZ A., González D. **“Sistemas de Detección de Intrusiones”**. Versión 1.01, Disponible en http://www.dgonzalez.net/pub/ids/, Julio 2003.

GONZÁLEZ D. **“Sistemas de Detección de Intrusiones”**. Versión 1.01, Disponible en http://www.dgonzalez.net/pub/ids/, Julio 2003.

HAYKIN, Simon. “**Redes Neurais: princípios e prática”**. Porto Alegre: Bookman, 2001.

HAYKIN Simon. **“Neural Networks: a Comprehensive Foundation”**. Macmillan, 1994.

HEADY, R et al. **“The architecture of a network level intrusion detection system”**. Technical report, Computer Science Department. University of Mexico. Agosto 1990.

HEBB D. (1949). **“The Organization of Behavior”**. Willey, New York, USA

HOLZ T., Meier M. y König H. **“High-Efficient Intrusion Detection Infraestructure”**. Infrastructure. DFN-Arbeitstagung über Kommunikationsnetze, 217-232, 2003.

HOTELLING, H. (1933) **“Analysis of a Complex of Statistical Variables into Principal Components”**. Journal of Education Psychology 24: 417-444.

IEEE, junio 2011, **“La Evolución del concepto de Seguridad”.** Documento Marco 05/2011. España.http://www.ieee.es/Galerias/fichero/docs_marco/2011/DIEEEM05-2011EvolucionConceptoSeguridad.pdf

INTECO, **“Desbordamiento de buffer”.** España. http://www.inteco.es/wikiAction/Seguridad/Observatorio/area_juridica_seguridad/Enciclopedia/Articulos_1/Desbordamiento_de_Buffer#definicion

INTECO-CERT, **“Noticias en la red”.** España. http://cert.inteco.es/cert/news/Noticias_CERT/?postAction=newsFilterTag&idTag=1001052440

INTERNET ENGINEERING TASK FORCE (IETF). **RFC 2828**, http://www.ietf.org/rfc.html

IRIS DATASET – **“UCI Machine Learning Repository”**. http://archive.ics.uci.edu/ml/datasets/Iris , 1996.

ISACA, November 2012. **COBIT-5-and-infosec.ppt.** http://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&ved=0CDkQFjAB&url=http%3A%2F%2Fwww.isaca.org%2FCOBIT%2FDocuments%2FCOBIT5-and-InfoSec.ppt&ei=5jJvUYK7L9GLhQejz4CYCA&usg=AFQjCNFXRhePL9moBnPC0KQTPvvykiV-dw&bvm=bv.45368065,d.ZG4

ISO/IEC 17799:2005, ISO (Organización Internacional de Normalización). http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=50297

ITOH, T., Takakura, H., Sawada, A., Koyamada, K. (2006) Hierarchical Visualization of Network Intrusion Detection Data. IEEE Computer Graphics and Applications 26(2): 40-47

JACKSON K. **“Intrusion Detection System Product Survey”**. Technical Report LA-UR-99-3883. Distributed Knowledge Systems Team, Computer Research and Applications Group, Los Alamos National Laboratory, New Mexico, Junio 1999.

KAYACIK, H. G., Zincir-Heywood, A. N., and Heywood, M. I. (2003). **“On the capability of an SOM based intrusion detection system”**. In Proceedings of the International Joint Conference on Neural Networks, volume 3, pages 1808–1813

KEIM, D.A. (2002). **“Information Visualization and Visual Data Mining”**. IEEE Transactions on Visualization and Computer Graphics 8(1): 1-8

KHALED, Labib. (2004). **Computer Security and Intrusion Detection.** Crossroads 11(1): 2-2

KEMMERER, R.A., Vigna, G. (2002) **Intrusion Detection: a Brief History and Overview**. Computer 35(4): 27-30.

LEI, J. Z. and GHORBANI, A. (2004). **“Network intrusion detection using an improved competitive learning neural network”**. In Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR), pages 190–197

LINDQVIST U. **“On the Fundamentals of Analysis and Detection of Computer Misuse”**. Thesis for the degree of Doctor of Philosophy, Chalmers University of Technology. Göteborg, Suecia, 1999.

LUNDIN E. **“Logging for Intrusion and Fraud Detection”**. Thesis for the degree of Doctor of Philosophy, Department of Computer Engineering, Chalmers University, Göteborg, Suecia, 2004.

MACHADO Renato B. “**Uma Abordagem de Detecção de Intrusão baseada em Sistemas Imunológicos Artificiais e Agentes Móveis”**. Florianópolis, 2005. (Mestrado Ciências da Computação) Universidade Federal de Santa Catarina, 2005.

MAFRA Paulo M. et al. **“POLVO-IIDS: un sistema de detección de intrusos inteligente basado en anomalías”.** In: Simponsio brasileño en seguridad de la información y de sistemas computacionales, 8. , 2008, Florianopolis. Anais..Florianopolis 2008. Disponibles en : http://sbseg2008.inf.ufrgs.br/proceedings/data/pdf/st02_02_artigo.pdf

MARCELO Antonio. **“Honeypots: a arte de iludir hackers”.** Rio de Janeiro: Brasport Livros e Multimídia Ltda, 2003.

MARKS D.G., Mell, P., Stinson, M. (2004) **“Optimizing the Scalability of Network Intrusion Detection Systems Using Mobile Agents”**. Journal of Network and Systems Manage-ment 12(1): 95-110.

MARTÍN Daniel F. **“Aplicação de Redes Neurais em IDS: comparação entre PortscanAI e SfPortscan”.** Novo Hamburgo, 2006. (Monografia Ciências da Computação) Centro Universitário Feevale, 2006.

MARTÍN DEL BRÍO B. y Sanz, A. (1997). **“Redes neuronales y sistemas borrosos”**. Madrid: Ra-Ma.

MARTY R. (2008). **“Applied Security Visualization”**. Addison-Wesley Professional

MCHUGH, J. (2001). “**Intrusion and Intrusion Detection”**. International Journal of Information Security 1(1): 14-35

PALMER Pol, Alfonso ; MONTAÑO Moreno, JJ**. “¿Que son las redes neuronales artificiales? Aplicaciones realizadas en el ámbito de las adicciones”**. Universidad de las Islas Baleares. Departamento de Psicología.

RICHARD S. Sutton and Andrew G. Barto. **“Reinforcement learning: An introduction”**. MIT Press,1998

SALATAS John (24 August 2011). **"Implementation of Competitive Learning Networks for WEKA"**. ICT Research Blog. Retrieved 28 January 2012.
http://jsalatas.ictpro.gr/implementation-of-competitive-learning-networks-for-weka/

SCATTERPLOTS Matrices.
http://stat.ethz.ch/R-manual/R-devel/library/graphics/html/pairs.html

SHELLY Xiaonan Wu, Wolfgang Banzhaf. **“The use of computational intelligence in intrusion detection systems: A review”.**Review Article. *Applied Soft Computing*, *Volume 10, Issue 1*, *January 2010*, *Pages 1-35.*

SHYU M., Chen, S., Sarinnapakorn, K., and Chang, L. (2003). **“A novel anomaly detection scheme based on principal component classifier”**. In Proceedings of ICDM’03, pages 172–179.

SPITZNER Lance, **“Know Your Enemy: Honeywall CDROM”**. http://www.honeynet.org. 2008

STANFORD, **“Stanford AI Lab - SAIL”**, Stanford Computer Science Department. http://ai.stanford.edu/.

STEFFEN JÚNIOR, Julio. “**Sistemas de Detecção de Intrusão”.** Nuevo Hamburgo, 2003. Centro Universitário Feevale.

TURBYFILL Carolyn, **“Risk-Based Metrics for Software system Design, Development, and Test”**. IEEE - November 2011. http://www.todaysengineer.org/2011/Nov/risk-management.asp

UTTS, Jessica M. **“*Seeing Through Statistics* 3rd Edition, Thomson**

Brooks/Cole", 2005, pp 166-167.

UMA, Bioestadística: Métodos y Aplicaciones U.D. Bioestadística. Facultad de Medicina. Universidad de Málaga.ISBN: 847496-653-1

WEISSHEIMER JÚNIOR, Carlos Alfredo. **"Hoynet: Um estudo Teórico e Prático"**. Nuevo Hamburgo, 2006. Centro Universitário Feevale.

WILKINSON, L. (2005). **"The Grammar of Graphics. Statistics and Computing Springer"**. New York

XIAOTONG Wang. **"The IDS model of intelligent design system"** Original Research Article. *Computers & Structures*, *Volume 61, Issue 3*, *November 1996*, *Pages 579-586*

XIANG C. and Lim S. M. (2005). **"Design of multiple-level hybrid classifier for intrusion detection system"**. In Proceedings of 2005 IEEE Workshop on Machine Learning for Signal Processing, pages 117–122.

ZANERO S. and Savaresi, S. M. (2004). **"Unsupervised learning techniques for an intrusion detection system"**. In Proceedings of the ACM symposium on Applied computing , pages 412–419, Nicosia, Cyprus

Anexos

ANEXO A

PRINCIPAL COMPONENT ANALYSIS (PCA)

```
function Statisticalpca(data, categories, types, m)
%%
%data: Conjunto de datos, donde cada fila es una muestra y cada columna una
variable
%categories: El tipo de ataque de cada que se produce en cada muestra
%types: Los distintos tipos de ataque que se producen, en formato Texto. Ejemplo:
%{'No';'Ataque 1';'Ataque 2'}; %Nombres de los diversos ataques (Ataque tipo 1, tipo
2, etc.)
%m: numero de proyecciones a generar en el scatter plot
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%        CAMBIAR TIPOS DE ATAQUES POR VALORES NUMERICOS              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

types={'Normal';'Botnet';'Escan1';'Escan2';'Dos1';'Dos2';'Remoto'};%Lista de los tipo
de ataque
colors={'b.';'g.';'r.';'c.';'m.';'y.';'k.'}; %Lista de colores con los que se ploteará cada tipo
de ataque
[a,b]=size(data); %Obtiene el número de filas (a) y de columnas (b) del conjunto
de datos inicial

%Cambia las etiquetas de texto de cada tipo de ataque por valores numéricos
%para ser usados en el ploteado
tnum=ones(a,1);
for i=1:a
   for j=1:size(types,1)
      if strcmp(categories{i,1},types{j,1})
         tnum(i,1)=j;
         break;
      end
   end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                          PREPROCESADO                                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Realiza el normalizado de los datos por columnas
for i=1:b
  maxl = max(data(:,i));
```

```
  if maxI ~= 0
     data(:,i)=data(:,i)/maxI;
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    ALGORITMO                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Se obtienen los autovalores (s) y autovectores (u) de la matriz de covarianzas del
%conjunto de datos (que es en lo que consiste el PCA)
[u,s,v]=svd(cov(data));

newdata=(data * u); %El nuevo conjunto de datos después de aplicar la
transformación PCA


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    SCATTER PLOT                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure('Name', 'Linear'); %Figura donde se ploteará el scatter plot

%Genera el scatter plot enfrentando las m primeras componentes (columnas)
%del conjunto de datos entre si
for i=1:m
   for j=1:m
      if i==j %La diagonal principal del Scatter plot no se plotea
         hold on;
         subplot(m,m,(m*(i-1))+j)
      else
         for k=1:size(types,1)%plotea cada clase con un color
            hold on;
            subplot(m,m,(m*(i-1))+j),
plot(newdata(tnum==k,i),newdata(tnum==k,j),colors{k,1});
            min1=min(min(newdata(:,i))); %Rescala los ejes al minimo y maximo en el
eje X e Y
            maxi1=max(max(newdata(:,i)));
            min2=min(min(newdata(:,j)));
            maxi2=max(max(newdata(:,j)));
            axis([min1,maxi1,min2,maxi2]);
         end
      end
   end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%                    UNA PROYECCION                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Plotea dos componentes XComp para el eje X, e YComp para el eje Y
figure();
XComp=2;
YComp=3;
for k=1:size(types,1)%cada tipo de ataque con un color distinto
   hold on;
   plot(newdata(tnum==k,XComp),newdata(tnum==k,YComp),colors{k,1});
end
   TITLE('2D PCA')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                          3D                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Plotea en 3D usando las columnas XComp, YComp, ZComp
XComp=1;
YComp=2;
ZComp=3;
S=10; %Tamaño de los puntos
figure();
scatter3(newdata(:,XComp),newdata(:,YComp),newdata(:,ZComp),S,tnum,'filled');
TITLE('ORIGINAL DATASET 3 VARIABLES')
```

ANEXO B

MAXIMUN LIKELIHOOD HEBBIAN LEARNING (MLHL)

```
function MLHL(data, categories,types,iters,lrate,m,p)
%%
%data: Conjunto de datos, donde cada fila es una muestra y cada columna una
variable
%categories: El tipo de ataque de cada que se produce en cada muestra
%types: Los distintos tipos de ataque que se producen, en formato Texto. Ejemplo:
%{'No';'Ataque 1';'Ataque 2'}; %Nombres de los diversos ataques (Ataque tipo 1, tipo
2, etc.)
%iters: Número de iteraciones
%lrate: el ratio de aprendizaje
%m: número de neuronas de salida
%p: parametro propio del CMLHL
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%        CAMBIAR TIPOS DE ATAQUES POR VALORES NUMERICOS              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
types={'Normal';'Botnet';'Escan1';'Escan2';'Dos1';'Dos2';'Remoto'};
colors={'b.';'g.';'r.';'c.';'m.';'y.';'k.'}; %Lista de colores con los que se ploteará cada tipo
de ataque
[a,b]=size(data); %Obtiene el número de filas (a) y de columnas (b) del conjunto de
datos inicial

%Cambia las etiquetas de texto de cada tipo de ataque por valores numéricos
%para ser usados en el ploteado
tnum=ones(a,1);
for i=1:a
   for j=1:size(types,1)
      if strcmp(categories{i,1},types{j,1})
         tnum(i,1)=j;
         break;
      end
   end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     PREPROCESADO                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Realiza el normalizado de los datos por columnas
for i=1:b
  maxl = max(data(:,i));
  if maxl ~= 0
     data(:,i)=data(:,i)/maxl;
  end
end

%Sphering de los datos
data=sphering(data,b);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    INICIALIZACIÓN                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Calcula el step e inicializa la matriz de pesos aleatoriamente
step=lrate/iters;
w=rand(m,b)*0.01;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     ALGORITMO                                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for j=1:iters
   tem=ceil(rand(1)*a);
   x=data(tem,:);
   y=w*x';
   e=x-y'*w;
   d=(lrate.*y)*((abs(e).^p).*(sign(e)));
   w=w+d;
   lrate=lrate-step;
 end
newdata=(data * w');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     SCATTER PLOT                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure('Name', 'Linear'); %Figura donde se ploteará el scatter plot

%Genera el scatter plot enfrentando las m primeras componentes (columnas)
%del conjunto de datos entre si
for i=1:m
```

```
   for j=1:m
      if i==j %La diagonal principal del Scatter plot no se plotea
         hold on;
         subplot(m,m,(m*(i-1))+j)
      else
         for k=1:size(types,1)%plotea cada clase con un color
            hold on;
            subplot(m,m,(m*(i-1))+j),
plot(newdata(tnum==k,i),newdata(tnum==k,j),colors{k,1});
            min1=min(min(newdata(:,i))); %Re-escala los ejes al mínimo y máximo en
el eje X e Y
            maxi1=max(max(newdata(:,i)));
            min2=min(min(newdata(:,j)));
            maxi2=max(max(newdata(:,j)));
            axis([min1,maxi1,min2,maxi2]);
         end
      end
   end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    UNA PROYECCION                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Plotea dos componentes XComp para el eje X, e YComp para el eje Y
figure();
XComp=2;
YComp=3;

for k=1:size(types,1)%cada tipo de ataque con un color distinto
   hold on;
   plot(newdata(tnum==k,XComp),newdata(tnum==k,YComp),colors{k,1});
end
   TITLE('2D PCA')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                       3D                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Plotea en 3D usando las columnas XComp, YComp, ZComp
XComp=1;
YComp=2;
ZComp=3;
S=10; %Tamaño de los puntos
figure();
scatter3(newdata(:,XComp),newdata(:,YComp),newdata(:,ZComp),S,tnum,'filled');TIT
LE('ORIGINAL DATASET 3 VARIABLES')
```

ANEXO C

COOPERATIVE MAXIMUN LIKELIHOOD HEBBIAN LEARNING (CMLHL)

```
function CMLHL(data, categories,types,iters,lrate,m,p,tau)
%%
%data: Conjunto de datos, donde cada fila es una muestra y cada columna una variable
%categories: El tipo de ataque de cada que se produce en cada muestra
%types: Los distintos tipos de ataque que se producen, en formato Texto. Ejemplo:
%{'No';'Ataque 1';'Ataque 2'}; %Nombres de los diversos ataques (Ataque tipo 1, tipo 2, etc.)
%iters: Número de iteraciones
%lrate: el ratio de aprendizaje
%m: número de neuronas de salida
%p: parametro propio del CMLHL
%tau: parámetro propio del CMLHL
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%          CAMBIAR TIPOS DE ATAQUES POR VALORES NUMERICOS                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

types={'Normal';'Botnet';'Escan1';'Escan2';'Dos1';'Dos2';'Remoto'};
colors={'b.';'g.';'r.';'c.';'m.';'y.';'k.'}; %Lista de colores con los que se ploteará cada tipo de ataque
[a,b]=size(data); %Obtiene el número de filas (a) y de columnas (b) del conjunto de datos inicial

%Cambia las etiquetas de texto de cada tipo de ataque por valores numéricos
%para ser usados en el ploteado
tnum=ones(a,1);
for i=1:a
   for j=1:size(types,1)
      if strcmp(categories{i,1},types{j,1})
         tnum(i,1)=j;
         break;
      end
   end
end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                       PREPROCESADO                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Realiza el normalizado de los datos por columnas
for i=1:b
```

```
   maxI = max(data(:,i));
   if maxI ~= 0
      data(:,i)=data(:,i)/maxI;
   end
end

%Sphering de los datos
data=sphering(data,b);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                   INICIALIZACIÓN                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Calcula el step e inicializa la matriz de pesos aleatoriamente
step=lrate/iters;
w=rand(m,b)*0.01;

%Inicializa las conexiones laterales
A=zeros(m,m);
for i=1:m
    for j=1:m
        A(i,j)=-cos(2*pi*(i-j)/m);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                    ALGORITMO                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Start CMLHL Algorithm
for j=1:iters
    tem=ceil(rand(1)*a);
    x=data(tem,:);
    y=w*x';

    %Apply lateral connections
    dy=1-A*y;
    y=y-tau*dy;

    e=x-y'*w;

    d=lrate*y*((abs(e).^p).*(sign(e)));
    w=w+d;
   lrate=lrate-step;
end
newdata=data*w';


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%                        SCATTER PLOT                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%f
igure('Name', 'Linear'); %Figura donde se ploteará el scatter plot

%Genera el scatter plot enfrentando las m primeras componentes (columnas)
%del conjunto de datos entre si
for i=1:m
   for j=1:m
      if i==j %La diagonal principal del Scatter plot no se plotea
         hold on;
         subplot(m,m,(m*(i-1))+j)
      else
         for k=1:size(types,1)%plotea cada clase con un color
            hold on;
            subplot(m,m,(m*(i-1))+j),
plot(newdata(tnum==k,i),newdata(tnum==k,j),colors{k,1});
            min1=min(min(newdata(:,i))); %Re-escala los ejes al mínimo y máximo en
el eje X e Y
            maxi1=max(max(newdata(:,i)));
            min2=min(min(newdata(:,j)));
            maxi2=max(max(newdata(:,j)));
            axis([min1,maxi1,min2,maxi2]);
         end
      end
   end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     UNA PROYECCION                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Plotea dos componentes XComp para el eje X, e YComp para el eje Y
figure();
XComp=2;
YComp=3;
for k=1:size(types,1)%cada tipo de ataque con un color distinto
   hold on;
   plot(newdata(tnum==k,XComp),newdata(tnum==k,YComp),colors{k,1});
end
   TITLE('2D PCA')



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                        3D                                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Plotea en 3D usando las columnas XComp, YComp, ZComp
XComp=1;
YComp=2;
ZComp=3;
S=10; %Tamaño de los puntos
figure();
scatter3(newdata(:,XComp),newdata(:,YComp),newdata(:,ZComp),S,tnum,'filled');
TITLE('ORIGINAL DATASET 3 VARIABLES')
```

ANEXO D

SPHERING

```
function [ndatos] = sphering (datos,tam)

[U,S,V]=svd(cov(datos));
ndatos=datos*V(:,1:tam);
[filas,columnas]=size(ndatos);

nb=diag(S);
for i=1:filas
    ndatos(i,:)=ndatos(i,:)./sqrt(nb(1:columnas))';
end
```

ANEXO E

PARSER2

```
function [newD]=parser2(text)
fil=size(text,1);
textJ=({text{:,1},text{:,2}})';
text3={text{:,3};}';
[f,c]=size(textJ);
numeric=zeros(f,1);
numeric3=zeros(fil,1);
k=1;

for i=1:size(textJ,1)
    if(numeric(i,1)==0)
        numeric(strcmp(textJ,textJ{i,1}),1)=k;
        k=k+1;
    end
end

k=1;

for i=1:size(text3,1)
    if(numeric3(i,1)==0)
        numeric3(strcmp(text3,text3{i,1}),1)=k;
        k=k+1;
    end
end

newD=[numeric(1:fil,1),numeric(fil+1:f,1),numeric3];
```

ANEXO F

ATTACKS

PRUEBA "ESCANEO DE LA RED

A) Escaneo de tipo red

nmap -sV -v 192.168.1.0/24

B) Escaneo agregando señuelos al proceso especificando determinados puertos (La IP de los señuelos en el análisis fueron: 192.168.1.37-39, "ME" se refiere a la IP de la víctima)

nmap -v -sS -D 192.168.1.37,192.168.1.39,ME 192.168.1.40 –p 21,22,23,80,8080,1434,445,161,1161,2161

PRUEBA "ATAQUE DOS"

A) Escenario1
Inundando de paquetes TCP SYN flags al host objetivo (192.168.1.34)

hping3 -V -c 200000 -d 120 -S -w 64 -p 80 -s 80 --flood --rand-source 192.168.1.34

B) Escenario2
Inundando de paquetes UDP flags al host objetivo en este caso será el 192.168.1.34

hping3 -V -c 200000 -d 120 -2 -w 64 -p 80 -s 80 --flood --rand-source 192.168.1.34

Nota adicional del ataque "DOS"

Para la realización de este ataque es necesario tener instalado hping3, de no ser así escribiendo el siguiente comando permitirá su instalación:

#apt-get install hping3

La IP Origen puede ser también spoofed (suplantar nuestra identidad por una falsa) con la opción: "-a"

La opción -1 representa al protocolo ICMP
La opción -2 representa al protocolo UDP

Para el protocolo TCP la diferenciación de los flags se da a continuación:

SYN = S
RST = R
ACK = A

Uno de los principales parámetros usados en hping3 están detallados en la siguiente tabla:

V	Verbose-mode
c	packet-count
d	data-size
S	SYN flag
w	win size
p	port
s	base source port

Tabla 4.2 Parámetros usados en hping3.

PRUEBA "BRUTE FORCE WITH HYDRA"

A) En este ataque esta direccionado al servicio ssh, donde para determinar el login hace uso del diccionario username.txt y el password del password.txt.

#./hydra -L username.txt -P password.txt -V 192.168.1.40 ssh